

UNIVERSIDADE DO EXTREMO SUL CATARINENSE - UNESC

CURSO DE CIÊNCIA DA COMPUTAÇÃO

MARCELO MILIOLI BRISTOT FILHO

**SCRATCH E O ENSINO DO PENSAMENTO COMPUTACIONAL APLICADO AOS
ESTUDANTES DO COLÉGIO UNESC**

CRICIÚMA

2020

MARCELO MILIOLI BRISTOT FILHO

**SCRATCH E O ENSINO DO PENSAMENTO COMPUTACIONAL APLICADO AOS
ESTUDANTES DO COLÉGIO UNESC**

Trabalho de Conclusão de Curso, apresentado
para obtenção do grau de Bacharel no curso de
Ciência da Computação da Universidade do
Extremo Sul Catarinense, UNESC.

Orientador: Prof. Me. Luciano Antunes

CRICIÚMA

2020

MARCELO MILIOLI BRISTOT FILHO

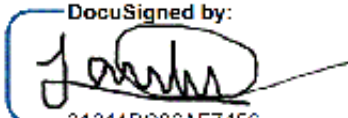
**SCRATCH E O ENSINO DO PENSAMENTO COMPUTACIONAL APLICADO AOS
ESTUDANTES DO COLÉGIO UNESC**

Trabalho de Conclusão de Curso aprovado pela Banca Examinadora para obtenção do Grau de Bacharel, no Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC, com Linha de Pesquisa em Informática na Educação.

Criciúma, 03 de agosto de 2020.

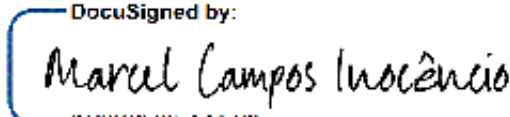
BANCA EXAMINADORA

DocuSigned by:



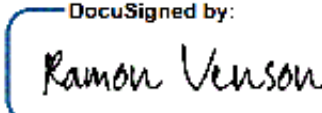
31311BC26AF7456
Prof. Luciano Antunes – Mestre – (UNESC) - Orientador

DocuSigned by:



2122C662EA4A42E
Prof. Marcel Campos Inocêncio – Especialista – (UNESC)

DocuSigned by:



540F55E5133044A
Prof. Ramon Venson – Mestre – (UNESC)

**A todos os que sonham com um futuro
melhor.**

AGRADECIMENTOS

Primeiramente a Deus, por tudo. Agradeço a minha família, principalmente meus pais e meu irmão, por todo o apoio e carinho nesta caminhada.

A todos os professores do curso, em especial ao professor Luciano Antunes pelas orientações que possibilitaram o desenvolvimento deste trabalho, e ao professor Kristian Madeira, pelo auxílio na elaboração do questionário.

Aos amigos do trabalho, que auxiliaram no projeto e permitiram que ele pudesse vir a ser colocado em prática.

Aos camaradas do curso de Ciência da Computação, que estiveram até o fim compartilhando os momentos na universidade.

Aos amigos do Rasga Canela, desde aqueles que jogam ocasionalmente, até os membros das formações das duas equipes que compõem a base da equipe.

Por fim, a todos que, direta ou indiretamente, contribuíram para moldar os rumos que me fizeram chegar até aqui, o meu muito obrigado.

“O gol é apenas um detalhe.”

Carlos Alberto Parreira

RESUMO

O pensamento computacional pode ser visto como uma ferramenta cognitiva com potencial para auxiliar no desenvolvimento do raciocínio lógico. Conceitos computacionais referentes a essa forma de pensar podem ser aplicados por meio do ensino de programação. O *Scratch* é uma plataforma digital utilizada com o intuito de criar projetos com o uso da programação, oferecendo uma abordagem visual através da utilização de blocos de encaixe simbolizando funções e conceitos. Durante este projeto, realizou-se a aplicação do *Scratch* em uma turma do 5º ano do Ensino Fundamental do Colégio UNESC, por meio de uma metodologia de ensino baseada nos princípios construtivistas que formam a base da plataforma, visando contribuir para o aprendizado de conceitos vinculados ao pensamento computacional. Por fim, foi aplicado um questionário aos participantes, procurando analisar o resultado da aplicação de tal ferramenta aos estudantes, observando se os fatores empregados contribuíram para o objetivo da pesquisa. Com a realização do projeto, pode-se concluir que a ferramenta *Scratch*, inserida no contexto da metodologia aplicada, contribuiu positivamente como estímulo para o aprendizado dos conceitos apresentados, fornecendo as bases para o desenvolvimento do raciocínio lógico baseado no pensamento computacional, além de proporcionar aos participantes um ambiente interessante e divertido, levando-os a interagir com a plataforma em um processo de aprendizagem ativa.

Palavras-chave: Pensamento computacional. *Scratch*. Programação em blocos. Informática na educação.

ABSTRACT

Computational Thinking can be seen as a cognitive tool with potential to help in the logical thinking development. Computational concepts related to this way of thinking can be applied through programming learning. *Scratch* is a digital platform used in order to create projects with programming, offering a visual approach through connected blocks that symbolize functions and concepts. During this project, *Scratch* was applied to a 5th year elementary school class at Colegio UNESC, through a teaching methodology based on the constructivist principles that form the basis of the platform, looking for the learning of computational thinking concepts. Lastly, a questionnaire was applied to the participants, analyzing the results of applying the tool to students, observing whether the factors employed helped to the research objective. With the project realization, it could be observed that *Scratch*, within the applied methodology, contributed positively as a stimulus to the concepts learning, providing the foundations for the computational-based logical thinking, besides giving an interesting and fun environment to the participants, leading them to interact with the platform in an active learning process.

Keywords: Computational thinking. *Scratch*. Blocks programming. Educational computing.

LISTA DE ILUSTRAÇÕES

Figura 1 – Pilares do pensamento computacional.....	20
Figura 2 – Conceitos e habilidades do pensamento computacional em diferentes literaturas	20
Figura 3 – Objetos de conhecimento do eixo pensamento computacional	22
Figura 4 – Ambiente <i>Scratch</i>	26
Figura 5 – Formatos diferentes de blocos	27
Figura 6 – Script de execução de movimentos.....	28
Figura 7 – <i>Script</i> de alteração de aparência.....	29
Figura 8 – Blocos de eventos	29
Figura 9 – <i>Script</i> de execução simultânea de dois atores	30
Figura 10 – <i>Script</i> de execução com atuação de sensor	30
Figura 11 – <i>Script</i> com utilização de operadores e variáveis.....	31
Figura 12 – Tutoriais disponibilizados pela plataforma <i>Scratch</i>	31
Figura 13 – Conceitos computacionais	40
Figura 14 – Criação de atividade durante aula (15/05).....	42
Figura 15 – Atividade final desenvolvida em aula (15/05)	42
Figura 16 – Demonstração de atividade realizada por aluno do Colégio UNESC.....	43
Figura 17 – Demonstração de atividade realizada por aluno do Colégio UNESC.....	44
Figura 18 – Apresentação dos projetos finais da pesquisa	45
Figura 19 – Apresentação dos projetos finais da pesquisa	45
Figura 20 – Perfil dos alunos – fator idade	46
Figura 21 – Perfil dos alunos – fator sexo	47
Figura 22 – Conhecimento prévio da ideia de programar em computadores.....	48
Figura 23 – Experiência prévia com programação em computadores	48
Figura 24 – Conceitos computacionais previamente conhecidos pelos participantes	49
Figura 25 – Dificuldade na compreensão do conteúdo.....	49
Figura 26 – Recepção positiva da forma utilizada pela pesquisa	50
Figura 27 – Utilização do <i>Scratch</i> fora da sala de aula.....	51
Figura 28 – Conceitos computacionais utilizáveis em outras aulas	52

LISTA DE TABELAS

Tabela 1 – Tipos de blocos do <i>Scratch</i>	27
--	----

LISTA DE ABREVIATURAS E SIGLAS

BNCC	Base Nacional Curricular Comum
CEP	Comitê de Ética em Pesquisa
CIEB	Centro de Inovação para a Educação Brasileira
CSBC	Congresso da Sociedade Brasileira de Computação
MIT	<i>Massachusetts Institute of Technology</i>
NRC	<i>National Research Council</i>
SBC	Sociedade Brasileira de Computação
TALE	Termo de Assentimento Livre e Esclarecido
TCLE	Termo de Consentimento Livre e Esclarecido
UNESC	Universidade do Extremo Sul Catarinense
VPL	<i>Virtual Programming Language</i>

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 OBJETIVO GERAL	15
1.2 OBJETIVOS ESPECÍFICOS	16
1.3 JUSTIFICATIVA	16
1.4 ESTRUTURA DO TRABALHO	17
2 PENSAMENTO COMPUTACIONAL	18
2.1 CARACTERÍSTICAS.....	19
2.2 O PENSAMENTO COMPUTACIONAL E A BASE NACIONAL CURRICULAR COMUM (BNCC)	21
2.2.1 Abstração	22
2.2.2 Algoritmos	22
2.2.3 Decomposição.....	23
2.2.4 Reconhecimento de padrões	23
3 PROGRAMAÇÃO EM BLOCOS	24
3.1 LOGO	25
3.2 <i>BLOCKY</i>	25
3.3 <i>SCRATCH</i>	25
3.3.1 <i>Scratch</i> em sala de aula e a metodologia construtivista.....	32
4 TRABALHOS CORRELATOS.....	33
4.1 A UTILIZAÇÃO DO <i>SCRATCH</i> COMO FERRAMENTA NO ENSINO DE PENSAMENTO COMPUTACIONAL PARA CRIANÇAS	33
4.2 ENSINO DE COMPUTAÇÃO COM <i>SCRATCH</i> NO ENSINO FUNDAMENTAL – UM ESTUDO DE CASO	34
4.3 ENSINO DE LÓGICA DE PROGRAMAÇÃO NO ENSINO FUNDAMENTAL UTILIZANDO O <i>SCRATCH</i> : UM RELATO DE EXPERIÊNCIA.....	34
4.4 LEARNING COMPUTER SCIENCE CONCEPTS WITH <i>SCRATCH</i>	35
5 A UTILIZAÇÃO DO AMBIENTE <i>SCRATCH</i> NO ENSINO DO PENSAMENTO COMPUTACIONAL	37
5.1 METODOLOGIA	37
5.1.1 Submissão do trabalho ao Comitê de Ética em Pesquisa (CEP)	38
5.1.2 Definição da metodologia de ensino	38
5.1.3 Planejamento do conteúdo a ser aplicado em aula	40
5.1.4 Execução das aulas	40

5.1.5 Análise dos resultados	46
5.2 RESULTADOS OBTIDOS.....	46
5.3 ANÁLISE DOS RESULTADOS	52
6 CONCLUSÃO	55
REFERÊNCIAS	57
APÊNDICE	61
APÊNDICE A – QUESTIONÁRIO APLICADO PARA AVALIAÇÃO DA PESQUISA	62
APÊNDICE B – ARTIGO	63
ANEXO	73
ANEXO A – APROVAÇÃO DO COMITÊ (CEP) PARA A REALIZAÇÃO DA PESQUISA	74

1 INTRODUÇÃO

A humanidade possui uma forma de observar o mundo intrinsecamente relacionada com a Computação, já que esta permite a utilização de metodologias, procedimentos e ferramentas para não só analisar e explicar o funcionamento de processos e entidades que integram o cotidiano, como também propor e desenvolver novos modelos e ideias que possibilitem a transformação da realidade na qual a mesma se insere (SBC, 2018).

Levando-se em consideração que a aplicação de tais práticas computacionais proporcionou profundas alterações sociais e tecnológicas, surge a necessidade de desenvolver uma melhor compreensão dos conceitos e fundamentos da Computação enquanto ciência, como forma de contextualizar as pessoas diante desse novo panorama encontrado (FRANÇA; TEDESCO, 2017).

O ensino dos conceitos de ciência da computação forma a base de uma habilidade conhecida como “pensamento computacional”. O pensamento computacional consiste na resolução de problemas e desenvolvimento de sistemas por meio de fundamentos como abstração e decomposição, podendo ser utilizável por todas as pessoas, por envolver o estudo do pensamento humano, incluindo uma grande quantidade de ferramentas mentais para a resolução eficiente de problemas de qualquer área (WING, 2006, tradução nossa).

Devido à importância dessa habilidade para o desenvolvimento do indivíduo, gera-se uma discussão acerca da possibilidade de promovê-la na educação básica, como forma de estimular esse pensamento desde os anos iniciais na formação do raciocínio dos estudantes. Segundo França et al. (2014), diversas ações já foram realizadas para disseminar o ensino do pensamento computacional, seja na inclusão direta na grade curricular, como também em alternativas que aproveitem da interdisciplinaridade da Computação.

Apesar de não haver consenso com relação à melhor metodologia a ser utilizada nessa missão, nota-se a existência de três pilares básicos para nortear tais atividades: abstração (extrair características importantes do problema), automação (substituição do trabalho manual) e análise (estudo dos resultados) (ANDRADE et al., 2013).

Uma das formas encontradas para o ensino desses conceitos consiste em utilizar ferramentas que situam o usuário no contexto da programação, por ser uma

aplicação prática dos fundamentos do pensamento computacional. Dentre essas ferramentas, pode-se destacar o *Scratch*.

O *Scratch* é um projeto desenvolvido pelo *Lifelong Kindergarten Group*, do *MIT Media Lab*, disponibilizado de forma gratuita para estimular o ensino de programação, utilizando o conceito de orientação a blocos de comando. Como sintetizado por Rusk, Resnick e Maloney (2009, apud FRANÇA; AMARAL, 2013), *Scratch* possibilita o desenvolvimento de habilidades de aprendizagem nas três áreas chaves descritas pela *Partnership for the 21st Century*: Habilidades de Informação e Comunicação; Habilidades de Pensamento e Resolução de Problemas; Habilidades Interpessoais e de Autodirecionamento.

A ferramenta *Scratch* utiliza uma abordagem visual, paradigma reconhecido por sua proeminência em designar o ensino de programação para jovens iniciantes no mundo da computação, ao gerar um ambiente de fácil utilização, codificando elementos gramaticais em blocos de ações interativos para o usuário (WEINTROP; WILENSKY, 2015, tradução nossa).

Pereira et al. (2012), ao aplicá-la na disciplina de Linguagem de Programação e Algoritmos, destaca que a utilização do *Scratch* pelos alunos auxilia na melhor compreensão de conceitos de programação de uma forma simples e divertida, permitindo o exercício da criatividade e do raciocínio científico lógico e preparando-os para a resolução de qualquer problema computacional proposto.

Arantes et al. (2014) denota a necessidade de formar os jovens rumo a uma “autonomia tecnológica”, onde, além de comportarem-se como usuários, possam compreender o funcionamento das tecnologias emergentes, como no ensino da programação, por exemplo.

Assim sendo, esta pesquisa teve como objetivo analisar a aplicação de uma metodologia que utilize a plataforma *Scratch* no ensino do pensamento computacional tendo como amostragem os estudantes do 5º (quinto) ano do Ensino Fundamental do Colégio UNESC.

1.1 OBJETIVO GERAL

Analisar a utilização da ferramenta *Scratch* em uma metodologia de ensino de pensamento computacional.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos consistem em:

- a) descrever o conceito de pensamento computacional;
- b) empregar a ferramenta *Scratch* como forma de desenvolver o aprendizado de raciocínio lógico;
- c) aplicar uma metodologia para avaliação do aprendizado da ferramenta empregada;
- d) aplicar a programação em bloco ao processo de ensino de programação no laboratório de Informática do Colégio UNESC nos estudantes do quinto ano do Ensino Fundamental;
- e) verificar se a programação em bloco contribuiu para o aprendizado do pensamento computacional dos estudantes, analisando os testes e critérios empregados.

1.3 JUSTIFICATIVA

O pensamento computacional, pela possibilidade de ser aplicado na resolução de problemas do cotidiano, permite ao indivíduo contextualizar-se diante do espaço em que se encontra e, no caso dos estudantes, consequentemente relacionar tal contexto com os conteúdos apresentados em sala de aula, como analisado em Gomes e Melo (2013).

A Computação nas escolas constitui o resultado de um processo natural de adaptação do ambiente escolar frente às mudanças encontradas na sociedade, visto que as ferramentas digitais possibilitaram a criação de um contexto completamente diferente do qual as escolas tradicionalmente estão habituadas (SBC, 2018).

Uma forma de estimular os estudantes nessa missão consiste no ensino simplificado dos conceitos de Computação, podendo, para isso, utilizar-se de ferramentas que estimulem a criatividade e o raciocínio crítico por meio de atividades que façam uso de programação orientada por blocos de comando, pois, como visto por Saito, Washizaki e Fukazawa (2017, tradução nossa), a utilização de conceitos visuais acaba sendo mais adequada para iniciantes não familiarizados com a programação.

O ambiente *Scratch*, uma das ferramentas que emprega tal paradigma, é utilizado em várias propostas de atividades envolvendo a modelagem de projetos, apresentando-se de forma satisfatória ao ser empregado no desenvolvimento de habilidades como o pensamento computacional (FRANÇA; AMARAL, 2013). Seu uso auxilia os estudantes no pensamento criativo, raciocínio crítico e no trabalho colaborativo (KALELIOĞLU; GULBAHAR, 2014, tradução nossa).

Segundo Oliveira (2014), torna-se fundamental que os conhecimentos básicos da Computação sejam adquiridos desde os primeiros anos da vida escolar do aluno, e várias experiências relatam de forma positiva a introdução de conceitos de programação como forma de promover o pensamento computacional em estudantes do Ensino Fundamental (AONO et al., 2017; KALELIOĞLU; GULBAHAR, 2014; OLIVEIRA et al., 2014; VON WANGENHEIM; NUNES; SANTOS, 2014).

A exposição dos conceitos lógicos para os alunos aplicarem posteriormente em atividades utilizando uma ferramenta visual mostra-se prática no estímulo à construção do conhecimento, estimulando métodos construtivistas em que se baseiam tais ferramentas, e o processo iterativo de desenvolvimento auxiliou no entendimento dos conceitos apresentados (AONO et al., 2017).

1.4 ESTRUTURA DO TRABALHO

Este projeto de pesquisa encontra-se dividido em seis capítulos, sendo o primeiro capítulo composto pela introdução, definição do problema, objetivo geral, objetivos específicos e a justificativa.

No capítulo 2 descreve-se o pensamento computacional, suas características e aplicações.

O terceiro capítulo aborda o conceito de Programação em Blocos, exemplos de ferramentas que utilizam seus princípios, destacando o *Scratch*, além da metodologia construtivista que serve de base para tal paradigma.

O capítulo 4 apresenta os trabalhos correlatos com a pesquisa.

O quinto descreve o trabalho realizado, sua metodologia, os resultados obtidos e a análise dos mesmos.

O sexto e último capítulo consiste na conclusão do trabalho.

2 PENSAMENTO COMPUTACIONAL

Para que o aprendizado do pensamento computacional possua um estímulo maior entre os participantes da pesquisa, e para que a mesma seja executada de forma clara e aprofundada, devem-se compreender os conceitos e os contextos que norteiam o pensamento computacional, sua origem, sua relação com habilidades extraídas da Computação e como pode ser efetuado seu ensino em salas de aula.

O conceito de “pensamento computacional” ganhou notoriedade em meados da primeira década de 2000, quando Jeanette Wing, por meio de um artigo publicado em 2006, intitulado “Computational Thinking” (“Pensamento Computacional”, em inglês), assinalou-o como uma habilidade fundamental para todas as pessoas, pela possibilidade de utilização em resoluções de problemas para além da área computacional. Porém, como apontado por Brackmann, nota-se que as ideias do pensamento computacional já existiam desde, pelo menos, a segunda metade do século XX, quando Papert e Solomon (1972) descreveram formas de resolução de problemas por meio de métodos computacionais, apesar de ainda não usarem a nomenclatura atual. O próprio Papert, posteriormente em 1980, chegou a utilizar tal expressão em novo artigo, apesar de não ganhar repercussão suficiente para gerar o desdobramento de novas pesquisas na época.

Apesar de não existir um consenso na definição precisa do termo “pensamento computacional”, mesmo nas comunidades educacionais e de ciência da computação, pode-se entendê-lo como uma “ferramenta cognitiva”, com aspectos inicialmente retirados da Computação (tais como abstração, automação, etc.) tendo a possibilidade de serem aplicados em outras áreas do conhecimento (USA NATIONAL RESEARCH COUNCIL, 2010, tradução nossa). Selby e Woollard (2013, tradução nossa) afirmam que denominadores em comum na definição do termo incluem a ideia de um “processo de pensamento”, além dos conceitos de abstração e decomposição.

Wing (2010, tradução nossa) definiu-o como o processo de pensamento envolvido na formulação de problemas e suas soluções, representadas de forma a serem realizadas efetivamente pelo agente processador de informação. Com isso, pode-se extrair que o pensamento computacional, ao consistir em um “processo de pensamento”, independe de tecnologias, sendo um conjunto de possíveis resoluções

de problemas, implicando em habilidades distintas (BOCCONI et al., 2016, tradução nossa).

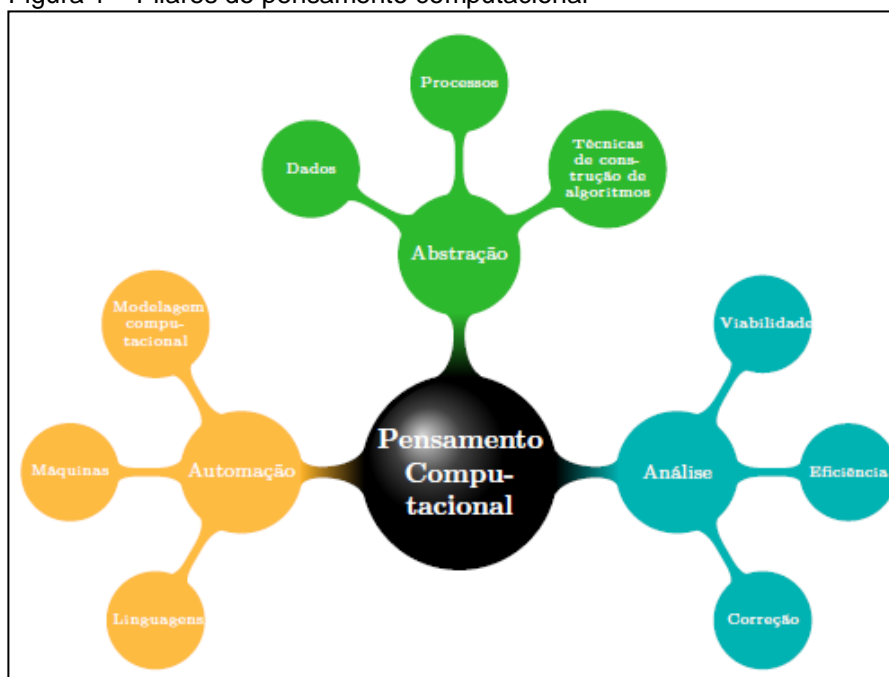
2.1 CARACTERÍSTICAS

Como forma de esclarecer o verdadeiro sentido referente ao que seria o pensamento computacional, Corrêa (2017) procurou delimitar suas características definindo-o pelo que ele não é, ou seja, procurando remover qualquer rotulagem que possa inicialmente ser definida sem aprofundamento do termo: “pensamento computacional” não está necessariamente relacionado com computadores, já que podem ser aplicados na solução de problemas de qualquer área; nem com a maneira com que os cientistas da computação pensam, sendo que tal método é utilizado apenas como base para compreender os conceitos referentes ao pensamento computacional, não o pensamento propriamente dito.

A Sociedade Brasileira de Computação (SBC), Sociedade Científica com finalidade de incentivar o progresso da área de Computação no Brasil, descreve o pensamento computacional como a “habilidade de compreender, definir, modelar, comparar, solucionar, automatizar e analisar problemas (e soluções) de forma metódica e sistemática”, sendo um dos eixos de conhecimento da área de Computação (SBC 2020).

Ribeiro et al. (2017) procurou delimitar as características do pensamento computacional em três pilares: abstração, consistindo nas técnicas utilizadas na construção de algoritmos, análise, referente ao consequente processo de verificação dos algoritmos desenvolvidos, e automação, onde ocorre a mecanização das soluções. A figura 1 representa tais pilares e suas ramificações.

Figura 1 – Pilares do pensamento computacional



Fonte: Ribeiro et al. (2017).

O Conselho Nacional de Pesquisa dos Estados Unidos (*National Research Council*, NRC), ao realizar em 2011 o “*Workshop sobre os Aspectos Pedagógicos do Pensamento Computacional*”, reportou a opinião de diversos atuantes na área da Computação com relação às características referentes ao pensamento computacional. A professora Janet Kolodner, por exemplo, caracterizou-o como um conjunto de habilidades interdisciplinares (NRC 2011, tradução nossa). Porém, assim como sua definição, as características do pensamento computacional variam de acordo com os artigos e pesquisadores. A figura 2 apresenta um compilado comparando alguns dos conceitos e habilidades encontrados em literaturas referentes ao assunto realizado por Bocconi et al. (2016, tradução nossa).

Figura 2 – Conceitos e habilidades do pensamento computacional em diferentes literaturas.

Barr e Stephenson (2011)	Lee et al (2011)	Grover e Pea (2013)	Selby e Woollard (2013)	Angeli et al (2016)
Abstrair	Abstrair	Abstrair	Abstrair	Abstrair
Algoritmos	Nenhum	Algoritmos	Algoritmos	Algoritmos
Automação	Automação	Nenhum	Nenhum	Nenhum
Nenhum	Analisar	Nenhum	Nenhum	Nenhum
Nenhum	Nenhum	Condicionar	Nenhum	Nenhum
Decompor	Nenhum	Decompor	Decompor	Decompor
Nenhum	Nenhum	Depurar	Nenhum	Depurar
Nenhum	Nenhum	Avaliar	Avaliar	Nenhum
Nenhum	Nenhum	Nenhum	Generalizar	Generalizar

Fonte: Bocconi et al. (2016, tradução nossa).

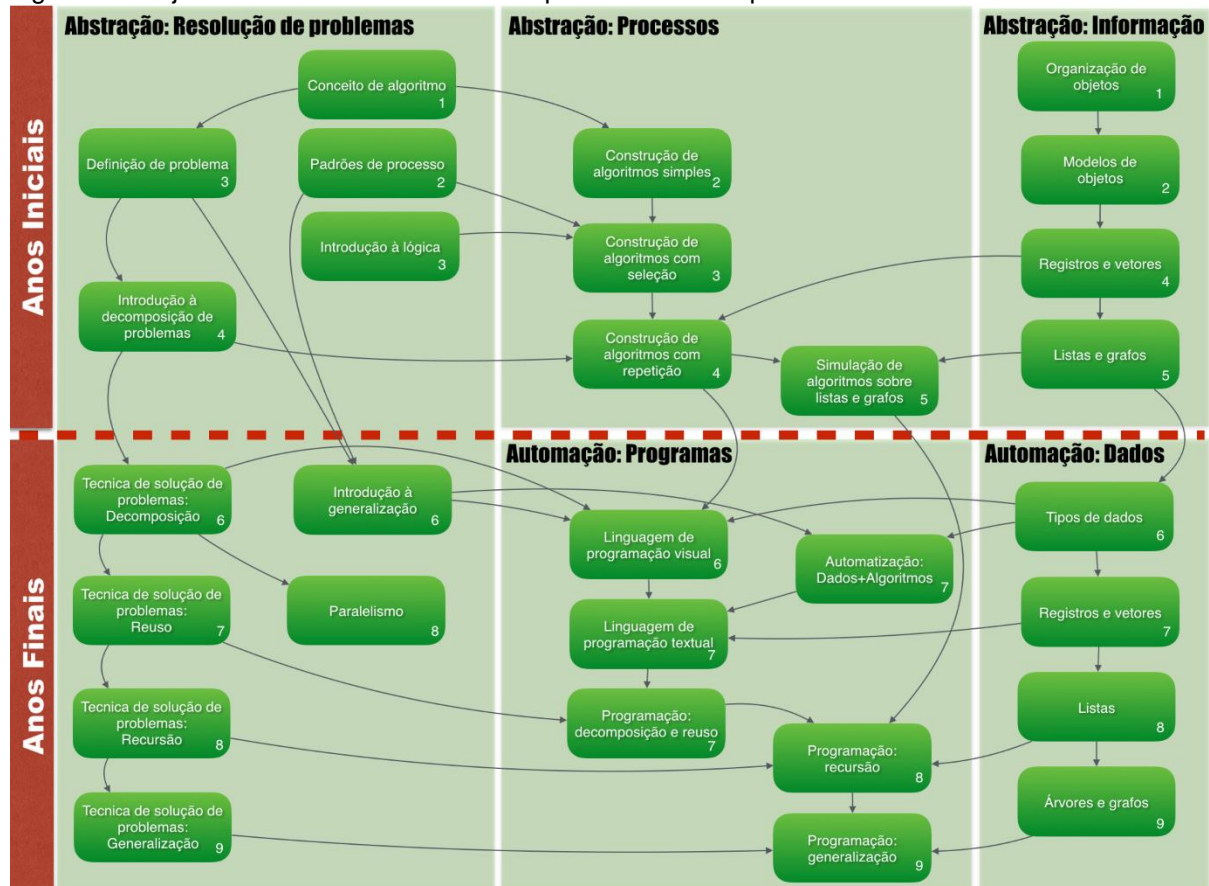
2.2 O PENSAMENTO COMPUTACIONAL E A BASE NACIONAL CURRICULAR COMUM (BNCC)

A Base Nacional Curricular Comum é o documento que regulariza os conteúdos a serem explorados para desenvolver o aprendizado dos estudantes da Educação Básica no país, estabelecendo conhecimentos, competências e habilidades que norteiam os sistemas de ensino em de todas as escolas do território nacional (BRASIL, 2016).

Reconhecendo a importância de incluir no currículo da Educação Básica o ensino de conceitos vinculados com a Computação, a SBC elaborou, em 2017, o documento intitulado “Referências de Formação em Computação: Educação Básica”, onde apresenta uma proposta de competências e habilidades a serem ensinadas nas escolas, como forma de contribuir no desenvolvimento da análise crítica e capacidade de solução de problemas dos estudantes (SBC, 2017). A figura 3 apresenta os objetos de conhecimento referentes ao eixo definido como “pensamento computacional”.

O Centro de Inovação para a Educação Brasileira (CIEB), baseando-se tanto nas referências de formação propostas pela SBC quanto nos parâmetros da BNCC, mais especificamente a quinta competência geral, que visa “compreender, utilizar e criar tecnologias digitais de informação e comunicação [...] para se comunicar, acessar e disseminar informações, produzir conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva” (BRASIL, 2016), desenvolveu um currículo referencial com diretrizes e orientações objetivando a construção de habilidades referentes aos temas de Tecnologia e Computação, dividindo-o em três eixos: cultura digital, tecnologia digital e pensamento computacional. Este último, por sua vez, sendo visto como “um dos pilares fundamentais do intelecto humano”, divide-se em quatro habilidades, a saber: Abstração, Algoritmos, Decomposição e Reconhecimento de Padrões (CIEB, 2019).

Figura 3 – Objetos de Conhecimento do eixo pensamento computacional



Fonte: Sociedade Brasileira de Computação (2017).

2.2.1 Abstração

Descrita por Wing (2008, tradução nossa) como a essência do pensamento computacional, a Abstração consiste em realizar a filtragem de dados, criando uma representação, ou ideia, do que se quer resolver, extraindo as informações relevantes e desconsiderando as demais (BRACKMANN, 2017).

Para o CIEB, o conceito de Abstração, ao filtrar os dados e promover sua classificação de forma a ignorar elementos que não são necessários, promove o auxílio na resolução de problemas, organizando as informações de forma estruturada.

2.2.2 Algoritmos

Como afirmado por Brackmann (2017), os algoritmos constituem a integração das outras habilidades, sendo formulado após utilizá-los diante do

processo de pensar computacionalmente. O método algorítmico consiste em uma série de passos ordenados que manipulam os tipos de dados abstratos para a solução de determinado problema (EASTEBROOK, 2014).

O currículo proposto pelo CIEB afirma que o conceito de Algoritmos é responsável por agregar todos os demais, sendo que o conjunto de instruções que consequentemente resulta na obtenção do objetivo definido pode ser desenvolvido por meio de diagramas ou em códigos, utilizando-se de uma linguagem de programação.

2.2.3 Decomposição

Ao realizar a decomposição de um determinado problema, facilita-se o processo de resolução, podendo realizar a análise das partes de uma forma mais atenta aos detalhes, identificando com mais nitidez suas funcionalidades (BRACKMANN, 2017). Ao dividir problemas complexos em atividades menores, pode-se gerenciar e resolver os mesmos de forma mais fácil (ARAÚJO et al., 2015).

O CIEB acrescenta que o processo de Decomposição permite que, após efetuar a análise separadamente, seja realizada a síntese das partes de forma que sua reconstituição possa gerar uma solução geral do problema.

2.2.4 Reconhecimento de Padrões

De acordo com Brackmann (2017), após a decomposição de um problema complexo, nota-se a existência de padrões (características compartilhadas) entre os subcomponentes gerados, podendo aplicar soluções genéricas para suas resoluções. Araújo et al. (2015) reconhecem o reconhecimento de padrões, ou generalização, como a identificação de partes funcionais em contextos específicos sendo reaproveitadas em situações mais amplas.

Desta forma, o currículo desenvolvido pelo CIEB evidencia a importância desse conceito a ser explorado diante do processo de resolução dos problemas.

3 PROGRAMAÇÃO EM BLOCOS

A programação orientada por blocos consiste em uma das variantes da programação visual, cujos ambientes promovem a utilização de peças, cada uma com determinada função, que encaixam entre si para formar os componentes do código, fazendo com que o usuário desenvolva seus programas apenas realizando a montagem dos blocos, recebendo o retorno visual (algumas vezes acompanhado por áudio) informando se a construção é válida (WEINTROP; WILENSKY 2015, tradução nossa).

Silva (2017) afirma que a ideia de utilizar blocos como forma de programar possui referências desde 1980, quando Seymour Papert coloca tal método para o ensino de programação para crianças. Papert baseou-se nos conceitos da filosofia construtivista do psicólogo suíço Jean Piaget, que afirma que a experiência pessoal da criança é responsável pela construção do conhecimento (CAMPOS, 2008), e criou, juntamente com a equipe de desenvolvimento do *Massachusetts Institute of Technology* (MIT), a ferramenta LOGO, uma das pioneiras no paradigma de programação em blocos, removendo o foco unicamente do código, além de proporcionar ao usuário uma liberdade maior para criação e representação de suas ideias.

Apesar de não ser um conceito novo, a programação em blocos tornou-se amplamente difundida recentemente, com o surgimento de uma nova geração de ferramentas, como *Scratch* (2007) e *Blockly* (2012), por exemplo. A crescente aparição de ferramentas que adotaram tal paradigma pode ser compreendida como uma forma de diminuir a barreira da programação em seus mais variados domínios de desenvolvimento (WEINTROP; WILENSKY, 2017, tradução nossa).

As linguagens de programação em blocos, por fazerem parte do grupo das *Visual Programming Languages* (VPLs), possuem certas vantagens quando relacionadas com as linguagens de texto. Saito, Washizaki e Fukazawa (2017, tradução nossa), por exemplo, ao realizarem a comparação dos dois paradigmas, concluíram que a utilização de conceitos visuais acaba sendo mais adequada para iniciantes não familiarizados com o mundo da programação.

3.1 LOGO

A linguagem LOGO foi desenvolvida em 1967 por Seymour Papert em parceria com outros colaboradores do MIT, com a intenção inicial de permitir que pessoas, mesmo em idade infantil, possam utilizar computadores como ferramenta de aprendizado. Seu componente geométrico, um cursor com a imagem de uma tartaruga, move-se na tela conforme as entradas realizadas pelo usuário, possibilitando a criação de figuras geométricas (PARDAMEAN; EVELIN; HONNI 2011, tradução nossa).

3.2 BLOCKY

A biblioteca *Blocky* foi produzida pela Google, com colaboração da equipe de desenvolvimento do laboratório *Media Lab* do MIT, e procura gerar um editor de programação visual para aplicações, utilizando-se de blocos encaixáveis para representar conceitos de programação (GOOGLE, 2016, tradução nossa). Os formatos dos blocos foram designados para facilitar a percepção das possibilidades de encaixe, auxiliando o usuário na construção de seus programas, sem a necessidade de preocupar-se com a sintaxe ou outras dificuldades impostas pelo desconhecimento de lógica de programação (SOUZA, 2018).

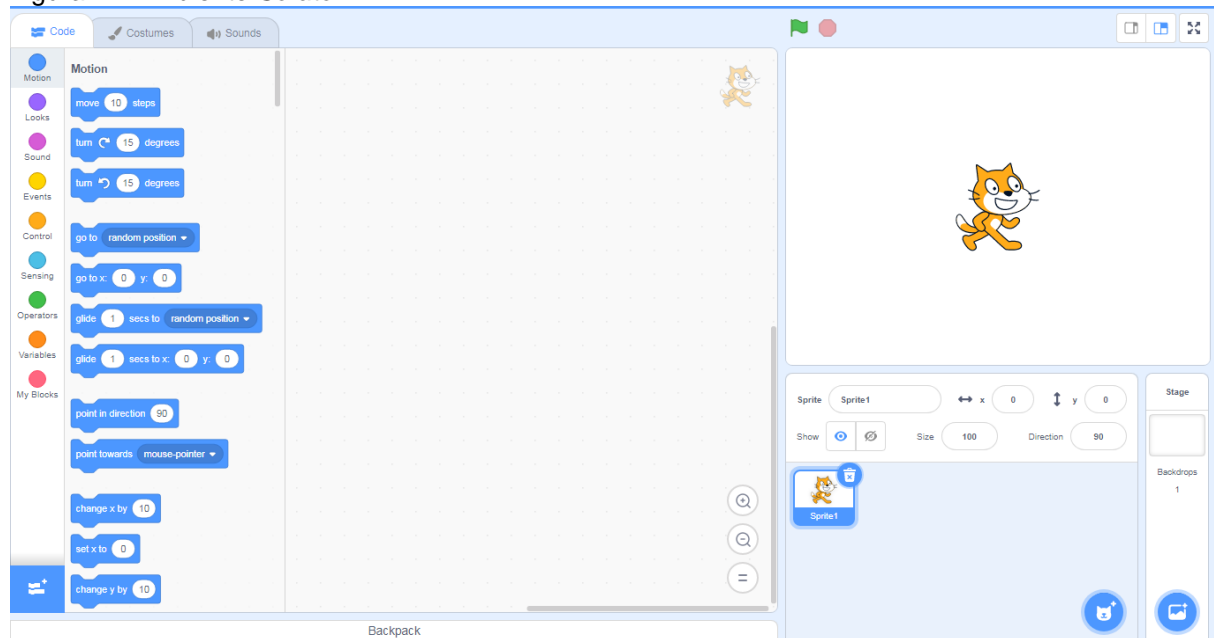
3.2 SCRATCH

A ferramenta *Scratch* consiste em um ambiente *web* gratuito de programação visual, projetado pelo grupo *Lifelong Kindergarten* no *Media Lab MIT*. Visando principalmente jovens entre 8 e 16 anos, atualmente é usado por pessoas de todas as idades (SCRATCH BRASIL, 2014).

Os programas desenvolvidos no *Scratch*, chamados de *scripts*, são criados arrastando e soltando blocos que representam os componentes do programa. Sua interface providencia o retorno imediato às ações do usuário, além de ignorar a questão da sintaxe (MEERBAUM-SALANT; ARMONI; BEN-ARI 2014, tradução nossa). Na figura 4, tem-se o ambiente de programação, onde se percebe o menu de seleção de peças. As peças azuis que aparecem na imagem, por

exemplo, são responsáveis pela função de movimentação (*Motion*) do personagem na tela à direita da figura.

Figura 4 – Ambiente Scratch









Fonte: Scratch (2019).

Os *scripts* são criados baseando-se no encaixe de blocos com representações visuais de itens relacionados à programação, com o próprio sistema realizando a filtragem das possibilidades de encaixe, já que os formatos dos blocos estão relacionados à sua disposição com relação aos outros comandos inseridos (MALONEY et al. 2010, tradução nossa).

A variação dos blocos pode ser percebida tanto na forma quanto na funcionalidade dos mesmos. Quanto às formas, existem seis tipos diferentes de blocos: *Hat*, *Stack*, *Boolean*, *Reporter*, *C* e *Cap*. Cada forma especifica um tipo de dado diferente que, por sua vez, atua na estrutura dos *scripts*, conectando-se entre si como peças de quebra-cabeças (SCRATCH, 2020). Essas formas foram desenvolvidas tendo em vista a prevenção de erros de sintaxe, já que somente determinados blocos podem realizar o encaixe específico, conforme visto na figura 5.

Figura 5 – Formatos diferentes de blocos

Forma	Nome	Função relacionada ao tipo de dado
	<i>Bloco Chapéu</i>	Iniciam os <i>scripts</i>
	<i>Bloco empilhável</i>	Realizam os comandos principais
	<i>Bloco booleano</i>	Representam as condições
	<i>Bloco repórter</i>	Representam os valores
	<i>Bloco em C</i>	Realizam os <i>loops</i>
	<i>Bloco final</i>	Finalizam os <i>scripts</i>

Fonte: *Scratch* (2020, tradução nossa).

Com relação às funções, os blocos dividem-se em dez grupos, denominados “categorias” ou “paletas”, diferenciado-os por cores como forma de facilitar a experiência do usuário com relação aos blocos que estão sendo utilizados (CASTRO, 2017). A tabela 1 mostra como essa divisão ocorre em diferentes categorias, com suas respectivas cores e quantidade de blocos.

Tabela 1 – Tipos de blocos do *Scratch*

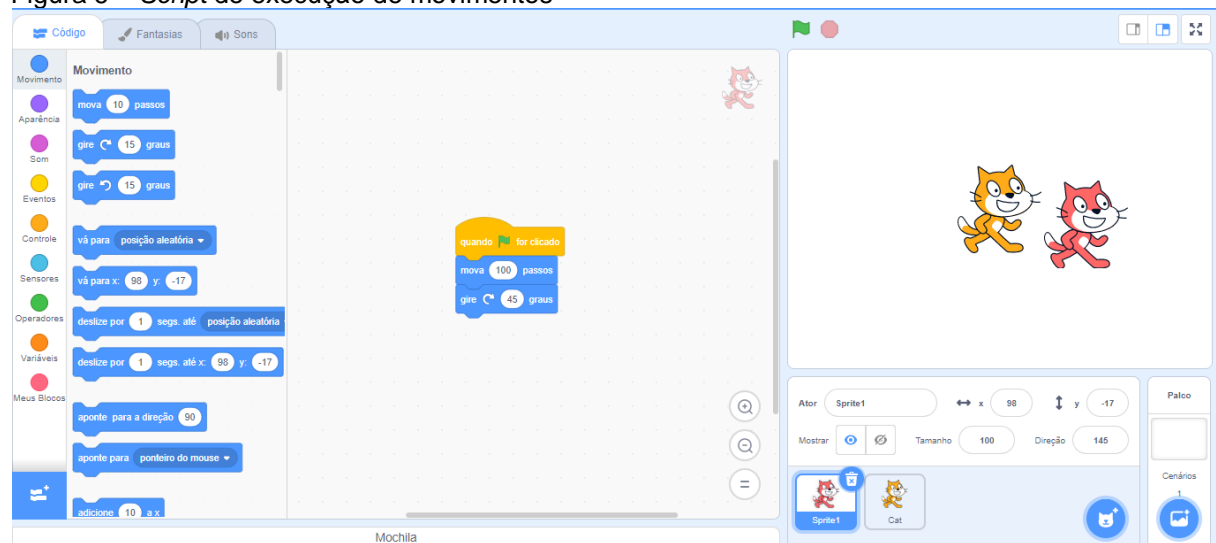
Categoria do bloco	Cor da categoria	Quantidade de blocos
Movimento	Azul médio	17
Aparência	Púrpura	23
Som	Magenta	16
Eventos	Amarela	8
Controle	Laranja (claro)	11
Sensores	Azul claro	21
Operadores	Verde	18
Variáveis(Variável/Lista)	Laranja (escuro)	17(5 variáveis/12 listas)
Meus blocos	Rosa	2

Fonte: *Scratch* (2020, tradução nossa).

A figura 6 demonstra como o processo de desenvolvimento dos *scripts* é realizado. Os blocos listados na tela à esquerda são arrastados para a tela central,

gerando as ações que ocorrerão quando o programa for iniciado. Baseando-se na criação de eventos (blocos em amarelo), o exemplo em questão utiliza ações vinculadas à categoria de movimento, a saber, movimentar determinada quantidade de passos e girar em um ângulo pré-determinado (blocos em azul médio), com o resultado final sendo apresentado na terceira tela. O personagem (gato) de coloração laranja representa a posição inicial do ator, enquanto o personagem vermelho foi utilizado para indicar seu novo posicionamento após a execução do *script* (clcando na bandeira verde).

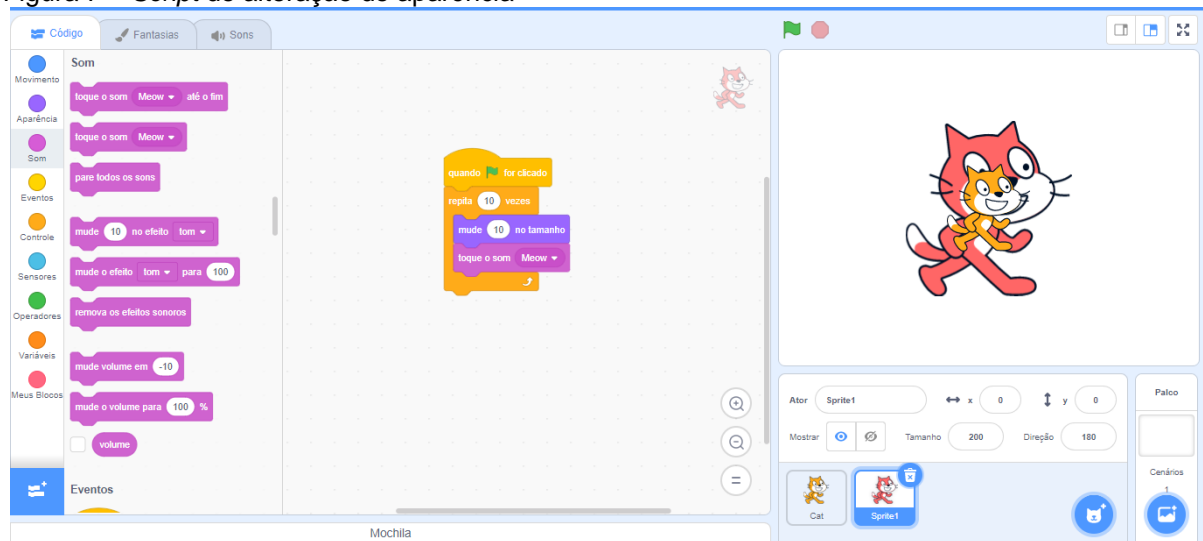
Figura 6 – *Script* de execução de movimentos



Fonte: Do autor.

Os blocos de controle, por sua vez, são adicionados para determinar o comportamento do programa, como estruturas de repetição e condicionais, podendo, por exemplo, permitir que as ações do programa continuem sendo executadas após sua inicialização, ao determinar a quantidade de execuções do comando. A figura 7 mostra a utilização do conceito de *loop*, ao inserir um bloco de repetição (Formato estilo “C”) com os comandos entre suas extremidades. Tais comandos pertencem à categoria de “Aparência” (modificar o tamanho, cor púrpura) e “Som” (reproduzir um som, cor magenta). Novamente, o “gato” em laranja representa o estado inicial do ator, enquanto o “gato” em vermelho o estado após a inicialização do *script*.

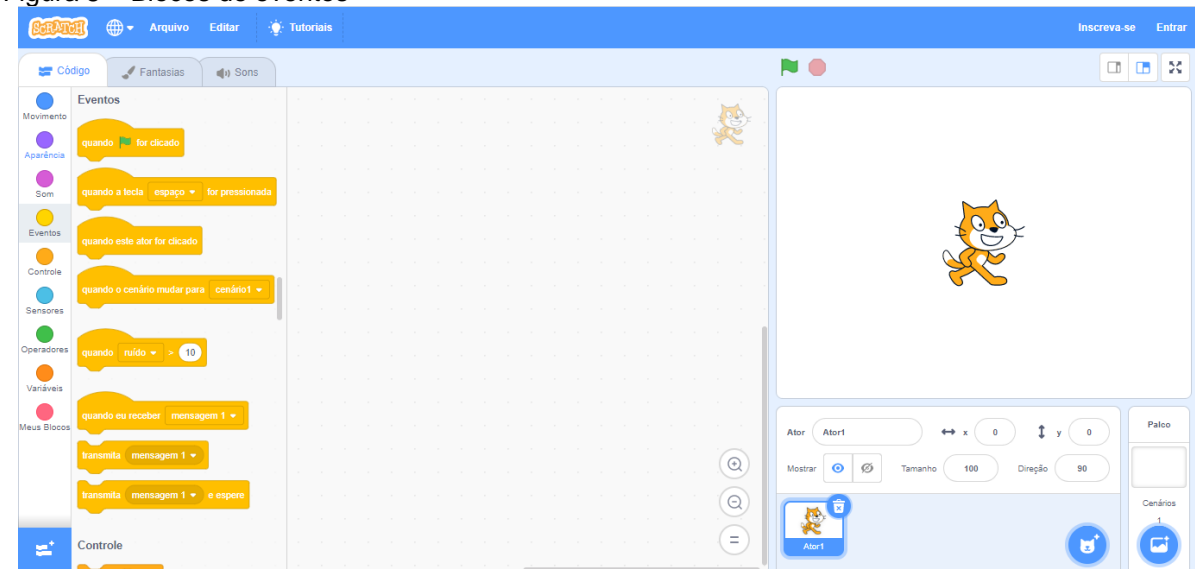
Figura 7 – Script de alteração de aparência



Fonte: Do autor.

Os blocos de eventos determinam o funcionamento do programa, ao definir as funções baseadas em comandos executados pelos usuários. A figura 8 mostra esses blocos, em sua maioria com formato *Hat*, por serem responsáveis por iniciar a execução dos blocos programados.

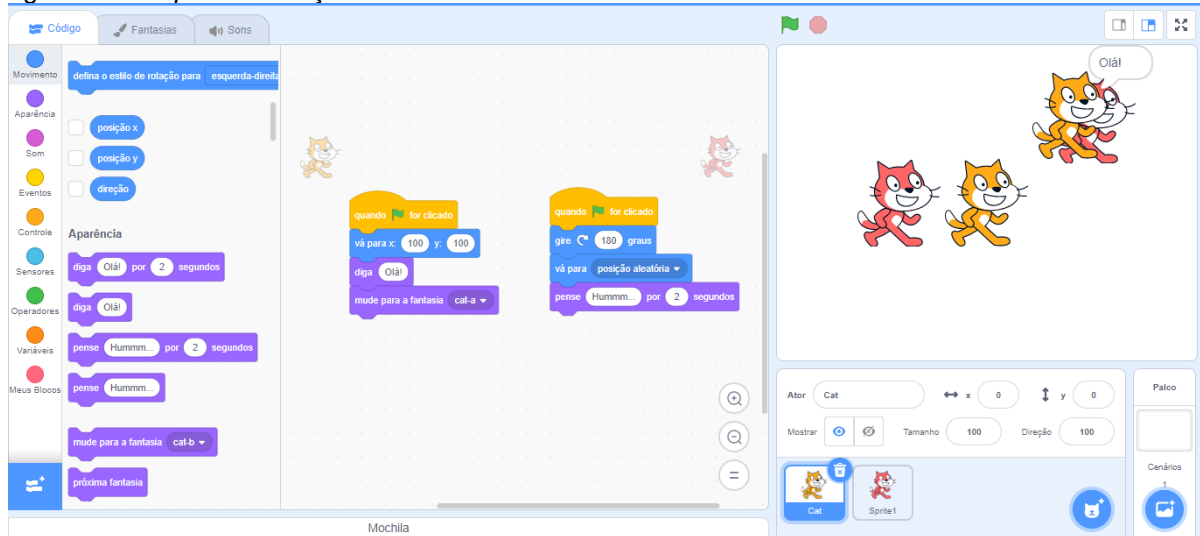
Figura 8 – Blocos de eventos



Fonte: Do autor.

Além disso, os blocos de eventos são adicionados de acordo com os atores e cenários selecionados, sendo que cada item possui determinadas ações que serão executadas individualmente após o início do programa. A figura 9 representa a sobreposição dos comandos para dois atores diferentes (os “gatos” laranja e vermelho) e seus respectivos estados pós-execução do programa.

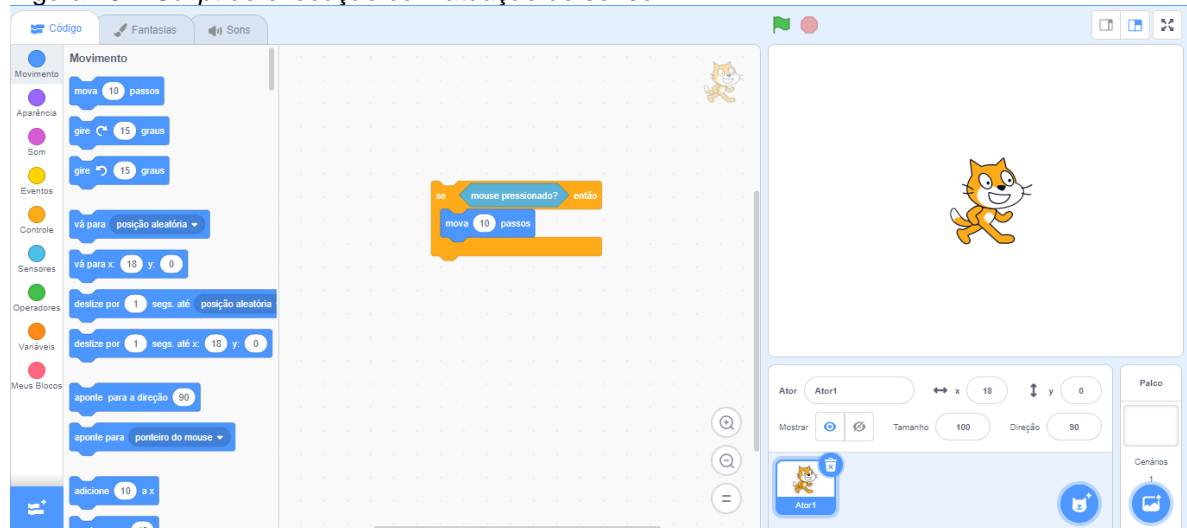
Figura 9 – Script de execução simultânea de dois atores



Fonte: Do autor.

Blocos de sensores (em azul claro) são utilizados para realizar a detecção de ocorrências no programa, como a variação nas coordenadas x e y do *mouse* ou se algum elemento está interagindo com o mesmo, por exemplo. Pode-se verificar no exemplo demonstrado pela figura 10 que haverá a movimentação de 10 passos do ator após o usuário pressionar o *mouse*.

Figura 10 – Script de execução com atuação de sensor

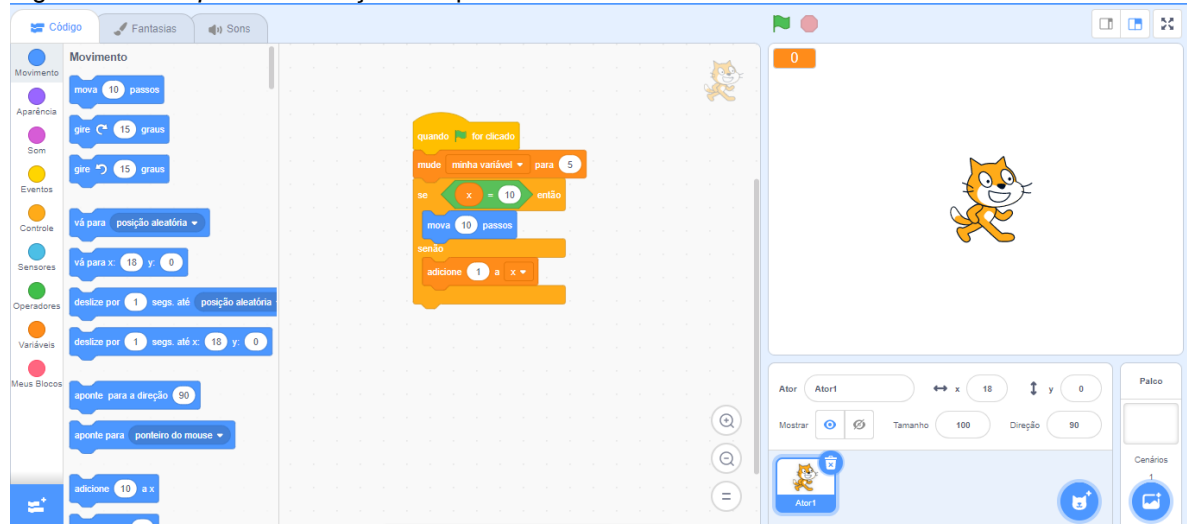


Fonte: Do autor.

Blocos de operações (em verde) e variáveis (laranja) também auxiliam no desenvolvimento dos *scripts*, pois permitem a utilização de operadores matemáticos que podem auxiliar na construção de estruturas condicionais, peças chave no

entendimento de algoritmos que estimulam o raciocínio computacional. Na figura 11 há a utilização de variáveis em um bloco representando o laço condicional criado por “*ifs*” e “*elses*” em estruturas de linguagens de programação de alto nível.

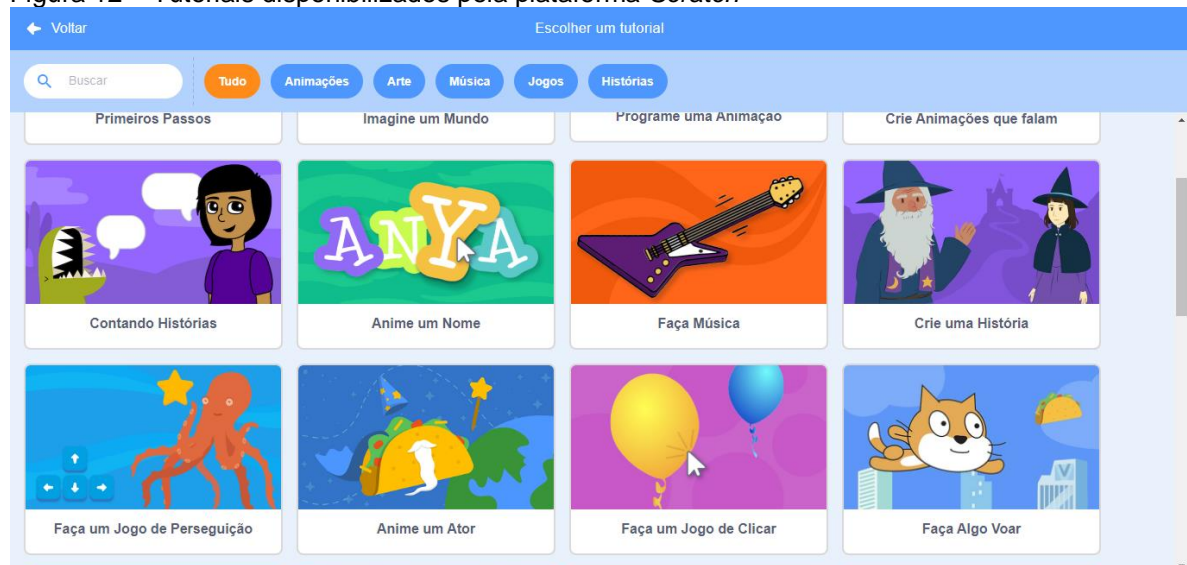
Figura 11 – Script com utilização de operadores e variáveis



Fonte: Do autor.

Outro ponto que favorece a experiência do usuário com a ferramenta é a customização de atores e ambientes, permitindo a construção de histórias interativas ou jogos personalizados de acordo com suas preferências. A figura 12 demonstra exemplos da própria plataforma que podem ser utilizados como tutoriais de aprendizagem.

Figura 12 – Tutoriais disponibilizados pela plataforma Scratch



Fonte: Do autor.

3.3.1 Scratch em sala de aula e a metodologia construtivista

A facilidade e a diversidade na utilização da ferramenta Scratch, aliados com a disponibilidade web e sua finalidade de promover a formação criativa das pessoas, permitiu sua popularização e, conseqüentemente, resultou no desenvolvimento de diversos projetos com objetivos educacionais, gerando uma vasta literatura tanto em termos quantitativos quanto em temática de ensino (Eloy et al., 2017). Ao realizar um levantamento bibliográfico observando as ferramentas utilizadas no Ensino-Aprendizagem do pensamento computacional, por exemplo, Bombasar et al. (2015) indicou o Scratch como a principal ferramenta utilizada nas propostas de ensino, com presença em 31 dos 106 estudos que relataram alguma prática de ensino-aprendizagem do pensamento computacional.

O uso dessa plataforma em sala de aula é potencializado pelo seu ideal pautado na abordagem construtivista, onde o aluno em questão promove a construção do conhecimento mediante interação com a ferramenta de ensino, em detrimento da abordagem tradicional, “instrucionista”, com o computador sendo inserido apenas como um recurso adicional ao ensino, com o estudante atuando como um mero expectador (GOMES, 2002).

A metodologia construtivista torna-se evidente em trabalhos que estimulam os participantes a criarem seus próprios projetos, utilizando o *Scratch* para o desenvolvimento de jogos e histórias interativas como produto final (CASTRO; KOSCIANSKI, 2017).

Além disso, ao considerar a construção do conhecimento como peça fundamental na estrutura dos trabalhos, acaba-se inferindo que os métodos de avaliação de aprendizagem também devem levar em conta tais aspectos, e, portanto, não podem estar limitados por objetivos terminais que impeçam a atividade criadora dos alunos (SOARES, 2002).

4 TRABALHOS CORRELATOS

Os próximos tópicos descrevem a utilização do Scratch como forma de promover o ensino de conceitos de Computação, sendo abordados, respectivamente, a criação de um jogo (AONO et al., 2017), o desenvolvimento de histórias interativas (VON WANGENHEIM; NUNES; SANTOS, 2014), Scratch como forma de auxiliar na aprendizagem de algoritmos (OLIVEIRA et al., 2014) e o desenvolvimento de materiais de ensino baseados na ferramenta (MEERBAUM-SALANT; ARMONI; BEN-ARI 2014, tradução nossa).

4.1 A UTILIZAÇÃO DO SCRATCH COMO FERRAMENTA NO ENSINO DE PENSAMENTO COMPUTACIONAL PARA CRIANÇAS

O artigo de Alexandre Hild Aono, Hugo Vianna Silva Rody, Daniela Leal Musa, Vanessa Andrade Pereira e Jurandy Almeida, A Utilização do Scratch como Ferramenta no Ensino de Pensamento Computacional para Crianças, foi publicado nos Anais do XXV Workshop sobre Educação em Computação realizado pela SBC no ano de 2017.

Nesse artigo, os autores utilizaram a ferramenta Scratch como uma forma de promover o ensino do pensamento computacional para estudantes do ensino fundamental. Visando alcançar tal objetivo, empregou-se uma metodologia expositiva aliada à utilização da ferramenta escolhida, ou seja, as aulas consistiam em explicações teóricas de determinado conceito relacionado ao tema “pensamento computacional”, seguidas por atividades práticas relacionadas aos tópicos recém-apresentados. Como agente motivador no processo de aquisição do conhecimento, propôs-se a criação de um jogo, desenvolvido progressivamente conforme os tópicos eram apresentados e as atividades realizadas em sala de aula (AONO et al., 2017).

Para verificar a abordagem descrita, realizou-se um estudo de caso, com a pesquisa sendo aplicada em 20 alunos do 6º ano do Ensino Fundamental de uma Escola Particular, possuindo média de idade entre 10 e 11 anos, no início do ano letivo de 2017. Como resultado, percebeu-se que houve a aquisição do conhecimento proposto pelos conteúdos ministrados nas aulas, e que a metodologia

aplicada contribuiu no processo de cativar a atenção dos participantes, com os conceitos sendo assimilados de forma fácil e agradável.

4.2 ENSINO DE COMPUTAÇÃO COM SCRATCH NO ENSINO FUNDAMENTAL – UM ESTUDO DE CASO

Christine Gresse von Wangenheim, Vinicius Rodrigues Nunes e Giovane Daniel dos Santos produziram o artigo Ensino de Computação com SCRATCH no Ensino Fundamental – Um Estudo de Caso, publicado no Volume 22, Número 3 da Revista Brasileira de Informática na Educação, de 2014.

No artigo, os autores apresentam um estudo de caso descrevendo o desenvolvimento de uma unidade instrucional (lições, exercícios ou atividades em torno de um tema logicamente sequenciadas) para promover o ensino de computação nos anos iniciais da Educação Básica, objetivando o envolvimento dos alunos na utilização do pensamento computacional como forma de resolução de problemas. Visando promover a interdisciplinaridade como definidora do conteúdo das aulas de computação, estabeleceu-se a criação de uma versão interativa de “Chapeuzinho Vermelho” como tema da unidade instrucional, utilizando o Scratch devido à facilidade e ampla utilização (VON WANGENHEIM; NUNES; SANTOS, 2014).

A pesquisa foi executada com 24 alunos de 6 a 7 anos do primeiro ano do Ensino Fundamental de uma escola privada de Florianópolis, no segundo semestre de 2013. O resultado desse estudo demonstra que, em se tratando da aplicação do pensamento computacional, todos os alunos conseguiram utilizar o ambiente de programação escolhido com facilidade, aplicando os conceitos de programação no desenvolvimento do projeto com o auxílio das características intuitivas da ferramenta.

4.3 ENSINO DE LÓGICA DE PROGRAMAÇÃO NO ENSINO FUNDAMENTAL UTILIZANDO O SCRATCH: UM RELATO DE EXPERIÊNCIA

O artigo de Millena Lauyse de Oliveira, Anderson Alves de Souza, Aline Ferreira Barbosa e Emanuel Francisco Spósito Barreiros, Ensino de lógica de programação no ensino fundamental utilizando o Scratch: um relato de experiência,

foi um projeto de extensão que culminou na publicação de seu relato no XXII Workshop sobre Educação nas Escolas, realizado no XXXIV Congresso da Sociedade Brasileira de Computação (CSBC), em 2014.

Os autores relatam a utilização da ferramenta Scratch como auxiliar na compreensão de algoritmos, por possibilitar um aprendizado mais intuitivo e visualmente agradável, além de estimular a criatividade e imaginação dos participantes (OLIVEIRA et al., 2014).

O projeto consistiu em um curso ministrado para 20 alunos do 9º ano do Ensino Fundamental de uma escola pública, onde o conteúdo de lógica de programação era ministrado com a possibilidade de criação de jogos e animações na ferramenta utilizada, com o estímulo aos alunos de desenvolverem suas atividades conforme seus interesses.

Os resultados obtidos foram otimistas, indicando que os alunos que participaram das atividades conseguiram compreender aspectos de computação, corroborando com o pensamento de que a computação possui importância no Ensino Básico, com a necessidade de, sempre que possível, inseri-la no currículo escolar.

4.4 LEARNING COMPUTER SCIENCE CONCEPTS WITH SCRATCH

Orni Meerbaum-Salant, Michal Armoni e Mordechai Ben-Ari produziram o artigo *Learning Computer Science Concepts with Scratch* (Aprendendo Conceitos de Ciência da Computação com Scratch, tradução nossa), tendo sua publicação realizada no volume 23 da revista britânica *Computer Science Education* de 2013.

Utilizando-se de métodos mistos de pesquisa, combinando técnicas quantitativas e qualitativas (como a realização de testes antes e depois da realização do estudo), metodologias e conceitos no estudo realizado, os autores descrevem no artigo o desenvolvimento de materiais de apoio baseados no Scratch, escolhido por sua ambientação bidimensional aparentar ser suficiente para alcançar as metas estipuladas (MEERBAUM-SALANT; ARMONI; BEM-ARI, 2013, tradução nossa).

A pesquisa foi realizada com duas classes, uma composta por 18 estudantes, enquanto outra por 28, ambas na faixa etária de 14-15 anos, cada uma recebendo uma aula de duas horas por semana por um semestre. Os dados obtidos

foram baseados em uma taxonomia desenvolvida pelos autores combinando outras duas existentes (Bloom e SOLO).

Os resultados mostraram-se positivos no que tange ao aprendizado de conceitos de computação pelos estudantes, principalmente aqueles que podem ser aplicados com poucas interações dentro do ambiente Scratch.

5 A UTILIZAÇÃO DO AMBIENTE SCRATCH NO ENSINO DO PENSAMENTO COMPUTACIONAL

O pensamento computacional, enquanto ferramenta cognitiva utilizada no processo de raciocínio lógico, pode ser um aliado na educação de crianças, ao contribuir positivamente no modo de pensar e resolver problemas, estimulando o desenvolvimento do pensamento infantil.

Ao ter suas bases extraídas da Computação, esse modo de pensar pode beneficiar-se de ferramentas e programas que utilizam os mesmos conceitos visando alguma finalidade específica, como o ensino de programação, por exemplo.

Ferramentas com didática voltada ao ensino de programação em blocos, como o *Scratch*, auxiliam no ensino de tais conceitos sem a necessidade de se ater a especificidades de linguagens, contribuindo para o desenvolvimento de atividades com foco na aprendizagem das ideias computacionais.

Ao utilizar ferramentas computacionais no ambiente educacional, o Colégio UNESC visa promover práticas pedagógicas inovadoras, ampliando a capacidade de aprendizagem do aluno em sala de aula (MENECH, 2014).

Diante desse cenário de modernização dos métodos e ferramentas de ensino, houve a proposta de contribuir para o ensino dos estudantes em fases iniciais com a execução de aulas para expor conceitos e práticas do pensamento computacional, como forma de aprimorar as habilidades com aplicação interdisciplinar.

Este trabalho consistiu em realizar aulas com os estudantes do 5º ano do Colégio UNESC por meio de uma plataforma digital de programação em blocos, como forma de analisar o ensino do pensamento computacional.

5.1 METODOLOGIA

Como forma de se alcançar os resultados, foram consideradas as seguintes etapas metodológicas: Levantamento bibliográfico; submissão do trabalho ao Comitê de Ética em Pesquisa (CEP); levantamento bibliográfico; definição da metodologia de ensino; planejamento do conteúdo a ser aplicado em aula; execução das aulas; análise dos resultados.

Durante a etapa de levantamento bibliográfico, compreenderam-se os assuntos referentes ao pensamento computacional, programação em blocos, a plataforma *Scratch*, sua aplicação no contexto de uma sala de aula e os conceitos de metodologia construtivista.

5.1.1 Submissão do trabalho ao Comitê de Ética em Pesquisa (CEP)

O Comitê de Ética em Pesquisa (CEP) é um órgão independente, obrigatório em instituições que realizam pesquisas envolvendo seres humanos, sendo responsável pela avaliação e acompanhamento dos aspectos éticos das mesmas (MUCCIOLI, 2004).

Devido à participação de crianças nesta pesquisa, tornou-se necessário submetê-la para análise e aprovação do Comitê de Ética em Pesquisa (CEP) da UNESCO, por meio do cadastro do projeto na ferramenta Plataforma Brasil (sistema que atua como base unificada das pesquisas realizadas em âmbito nacional).

Dois documentos imprescindíveis para a realização do projeto solicitados pelo CEP consistiram nos modelos do Termo de Assentimento Livre e Esclarecido (TALE, destinado aos participantes do projeto) e Termo de Consentimento Livre e Esclarecido (TCLE, destinado aos responsáveis pelos participantes, por serem menores de 18 anos), onde as informações da pesquisa eram descritas para posteriormente serem entregues aos seus envolvidos para aprovação escrita.

Ocorreu a aprovação do projeto conforme parecer número 3.889.080 (Anexo A).

5.1.2 Definição da metodologia de ensino

Após a aprovação do projeto pelo Comitê, seguiu-se com o planejamento das aulas. Inicialmente, entrou-se em acordo com o Colégio UNESCO para a realização das mesmas nas próprias instalações da instituição em horário de aula, utilizando o período referente à matéria de informática como forma de integrar o projeto à proposta pedagógica do colégio.

Visando alinhar o projeto com o formato das aulas realizadas com os alunos, optou-se por manter o padrão de utilização da sala com uma abordagem expositiva dos conteúdos, seguida por realização de atividades para aplicação

prática dos tópicos apresentados. O tempo disponibilizado para a execução do projeto (uma hora por semana, toda sexta-feira, durante seis semanas) contribuiu para um planejamento mais distribuído com relação à apresentação dos conceitos em contraste com a utilização de tempo.

Para determinar a maneira de utilização da ferramenta *Scratch*, foi levada em consideração a abordagem construtivista com a qual ela foi idealizada. Ao interpretar a criança como sujeito ativo na construção de seu conhecimento, essa forma de aprendizagem se beneficia de atividades em que os alunos participam efetivamente, adquirindo experiência ao manipular ferramentas que fornecem a solidez para o aprendizado (JONASSEN, 1996).

Desse modo, optou-se por determinar a entrega de projetos desenvolvidos por cada aluno, ou seja, de maneira individualizada, como proposta final das aulas realizadas nesta pesquisa. Com a liberdade de criação possibilitada aos usuários, o tema para a atividade proposta foi livre, tanto em forma (jogos, histórias, etc.) quanto em conteúdo, procurando promover a criatividade dos participantes ao aplicar os conceitos observados nas aulas em assuntos que são de seu interesse, contribuindo de forma positiva para o aprendizado no decorrer dos encontros.

Como forma de avaliar o desempenho final dos participantes, bem como observar a eficácia da pesquisa sob a perspectiva dos próprios alunos, definiu-se que ocorreria a aplicação de um questionário ao final do projeto, para que os estudantes respondessem de acordo com seu aprendizado e participação em sala de aula.

A criação do questionário baseou-se nas perguntas desenvolvidas por Pereira e Franco (2018), ao levarem em consideração os pontos elencados por Savi (2011) para modelar a avaliação da aprendizagem de uma metodologia de ensino de pensamento computacional. O questionário encontra-se no Apêndice A.

Com isso, ficou acertado o seguinte cronograma para a execução das aulas:

- a) a primeira aula seria essencialmente introdutória, consistindo na apresentação e descrição da finalidade do projeto, além da primeira demonstração da ferramenta;
- b) os dois encontros seguintes consistiriam na apresentação de conteúdos por meio de tutoriais da plataforma, com os estudantes

- replicando as atividades visualizadas;
- c) as duas próximas aulas seriam dedicadas aos alunos para realizarem o desenvolvimento dos projetos;
- d) o último encontro seria dedicado às apresentações dos projetos e preenchimento dos questionários.

5.1.3 Planejamento do conteúdo a ser aplicado em aula

Com a metodologia de ensino definida, a etapa seguinte consistiu em realizar o planejamento das aulas, levando em consideração a abordagem e o conteúdo a serem aplicados em sala de aula.

Os conceitos elencados para servir de base para a apresentação do projeto foram definidos de acordo com o guia “*Creative Computing – a design-based introduction to computational thinking*”, uma publicação escrita pelo grupo *ScratchEd* em 2011 como forma de prover um currículo para auxiliar professores no desenvolvimento do pensamento computacional com base na ferramenta Scratch. Os Conceitos Computacionais descritos pelo guia estão explanados na figura 13.

Figura 13 – Conceitos Computacionais

Conceito	Descrição
Sequência	Identificar uma série de passos para uma tarefa
Repetições	Realizar a mesma sequência várias vezes
Paralelismo	Promover a ocorrência simultânea das coisas
Eventos	Observar algo causando a ocorrência de outro
Condicionais	Tomar decisões baseadas em condições
Operadores	Apoio para expressões lógicas e matemáticas
Dados	Armazenar, obter e atualizar valores

Fonte: Brennan et al. (2011, tradução nossa).

5.1.4 Execução das aulas

A execução das aulas, prevista inicialmente para ter seu início durante o mês de abril, acabou sendo adiada devido às medidas restritivas adotadas pelo Colégio em virtude do surto de coronavírus, como o cancelamento imediato das

aulas presenciais. Esta imprevisibilidade gerou a necessidade de revisar o planejamento, propondo alterações no cronograma de execução.

A retomada das aulas ocorreu de forma virtual, utilizando a plataforma *Google Meet* para a realização dos encontros. Como forma de manter a mesma linha de proposta, adequando a pesquisa ao método escolar, a mesma ferramenta seria utilizada para mediar as aulas com os estudantes.

Com a previsão inicial de volta às aulas para o dia 01/06, o novo cronograma estabeleceu a realização das aulas com duração de uma hora entre as sextas-feiras dos dias 08/05 e 12/06, com os encontros anteriores à data de retorno (08, 15, 22 e 29 de maio) sendo realizados por meio da plataforma virtual *Google Meet*, enquanto os posteriores (05 e 12 de junho) inicialmente presenciais, aproveitando-se da volta à normalidade do Colégio.

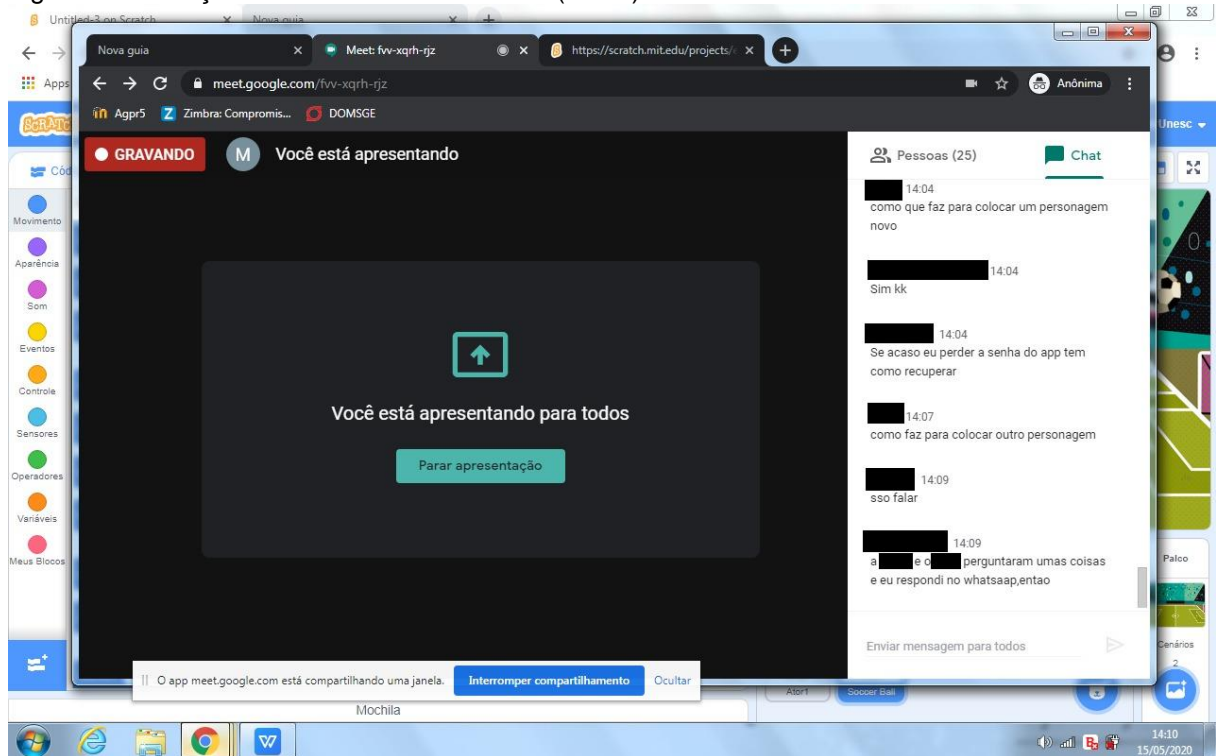
Para manter o protocolo ético na aplicação de pesquisa com crianças, foram encaminhados aos responsáveis *e-mails* com a explicação de como seriam realizadas as aulas do projeto, com uma cópia em anexo dos documentos com os termos de aceite, TCLE e TALE, para assinatura e posterior recolhimento.

A primeira aula (08/05) foi reservada para a apresentação e descrição da finalidade do projeto, familiarizando os participantes com a ferramenta e expondo o objetivo da realização da pesquisa. Após breve resumo explicando o cronograma das aulas, expôs-se aos alunos a proposta de realização de um projeto final para apresentação, tendo recepção positiva dos participantes. Para iniciar os trabalhos com a plataforma, o primeiro passo foi o cadastro de um usuário teste que seria utilizado pelo pesquisador durante a pesquisa, explicando aos participantes as etapas para que também pudessem realizar o registro para salvar os trabalhos. Até o final do encontro, foram discutidas pequenas funcionalidades no *Scratch* como forma de situar os estudantes nos primeiros passos na utilização da ferramenta, sem, entretanto, iniciar a apresentação de conceitos computacionais de fato.

A aula seguinte (15/05) foi destinada ao desenvolvimento conjunto de atividades que serviriam de referência para a elaboração dos projetos por parte dos estudantes, onde o professor utilizaria tutoriais disponibilizados na própria plataforma para que os alunos posteriormente replicassem de acordo com o conteúdo apresentado. Para situar os estudantes da relação entre os blocos de programação com seus conceitos referenciais, cada item adicionado era detalhado conforme seu respectivo equivalente computacional (descrição do conceito). As figuras 14 e 15

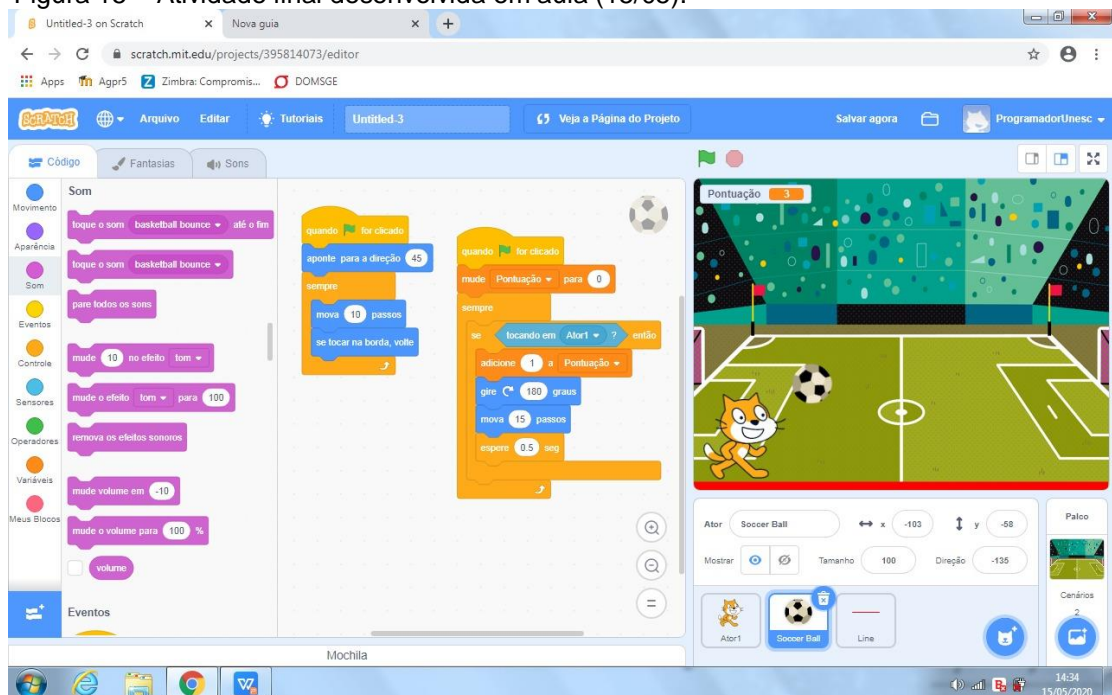
ilustram, respectivamente, o processo de criação de uma adaptação de um desses tutoriais sendo criada durante a aula e o resultado final do desenvolvimento.

Figura 14 – Criação de atividade durante aula (15/05).



Fonte: Do autor.

Figura 15 – Atividade final desenvolvida em aula (15/05).



Fonte: Do autor.

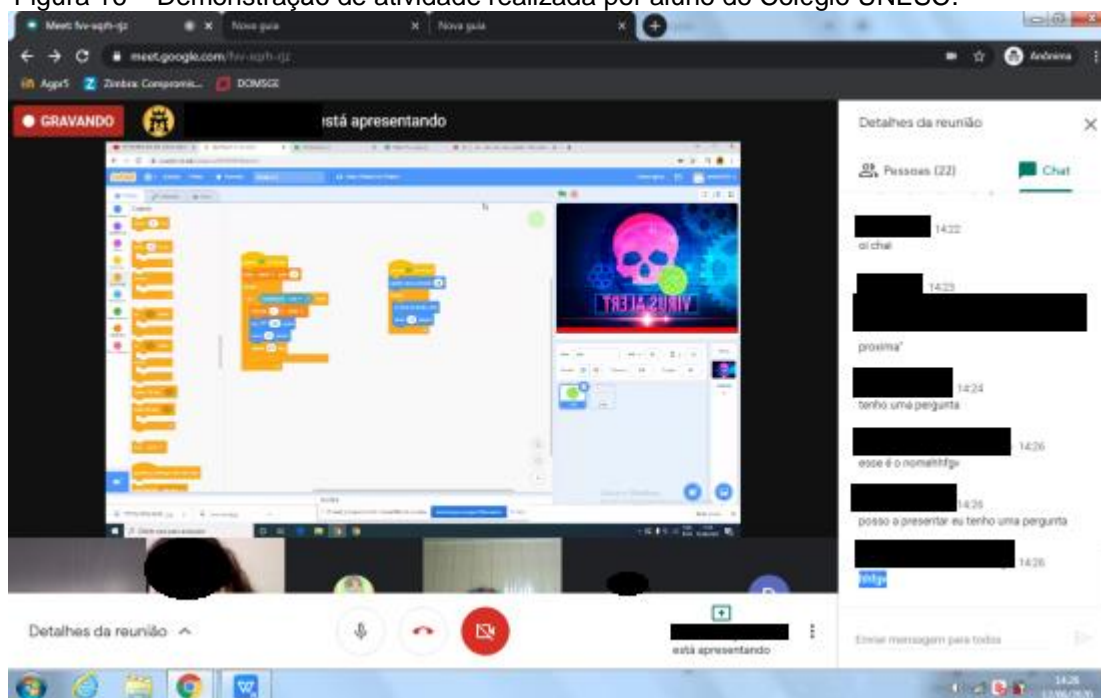
O encontro do dia 22/05, inicialmente programado para continuar a proposta de desenvolvimento de atividades para replicação, acabou não ocorrendo devido ao cancelamento das aulas na semana, sendo necessário prorrogar por mais uma semana o projeto, acertando com a professora responsável pela turma mais uma sexta-feira (19/06).

Na sexta-feira seguinte (29/05), houve a continuidade do processo de criação de atividades para exposição dos conceitos apresentados conforme aula anterior.

Apesar do retorno às aulas estar inicialmente previsto para 01/06, houve a prorrogação das restrições, impedindo assim a utilização dos ambientes físicos do Colégio UNESC. Com isso, manteve-se a utilização dos meios digitais durante a pesquisa.

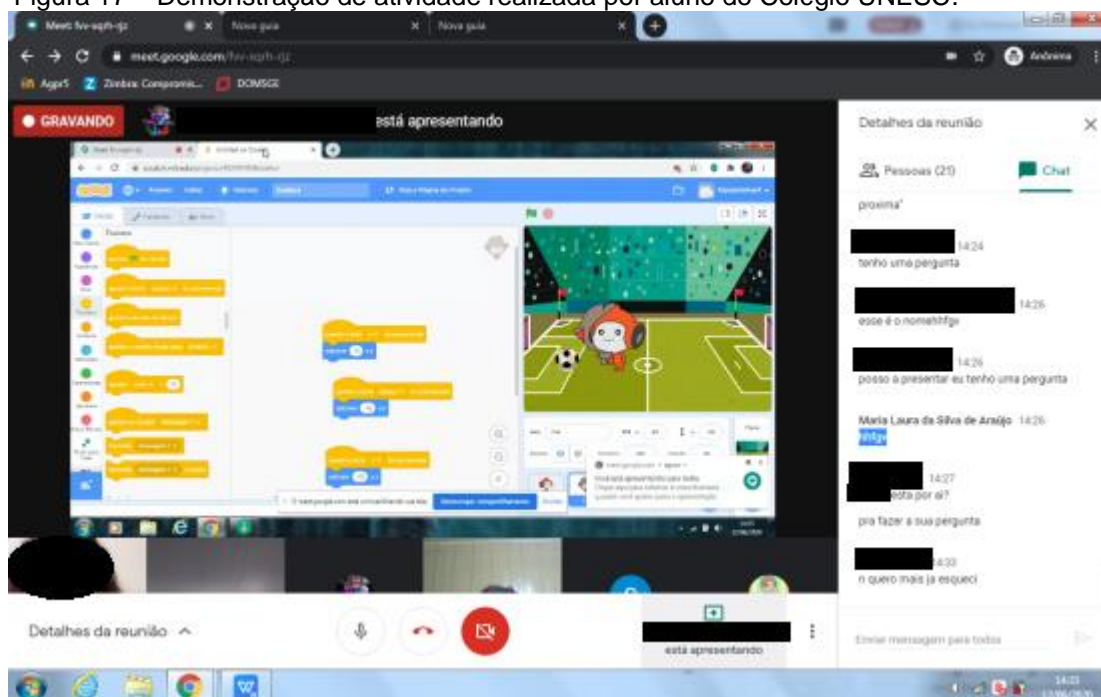
As aulas dos dias 05/06 e 12/06 foram dedicadas ao início do desenvolvimento das atividades dos participantes, com o professor auxiliando nas dúvidas conforme o andamento do encontro. Nas figuras 16 e 17 é possível observar a demonstração de projetos em processo de criação por parte dos estudantes.

Figura 16 – Demonstração de atividade realizada por aluno do Colégio UNESC.



Fonte: Do autor.

Figura 17 – Demonstração de atividade realizada por aluno do Colégio UNESC.

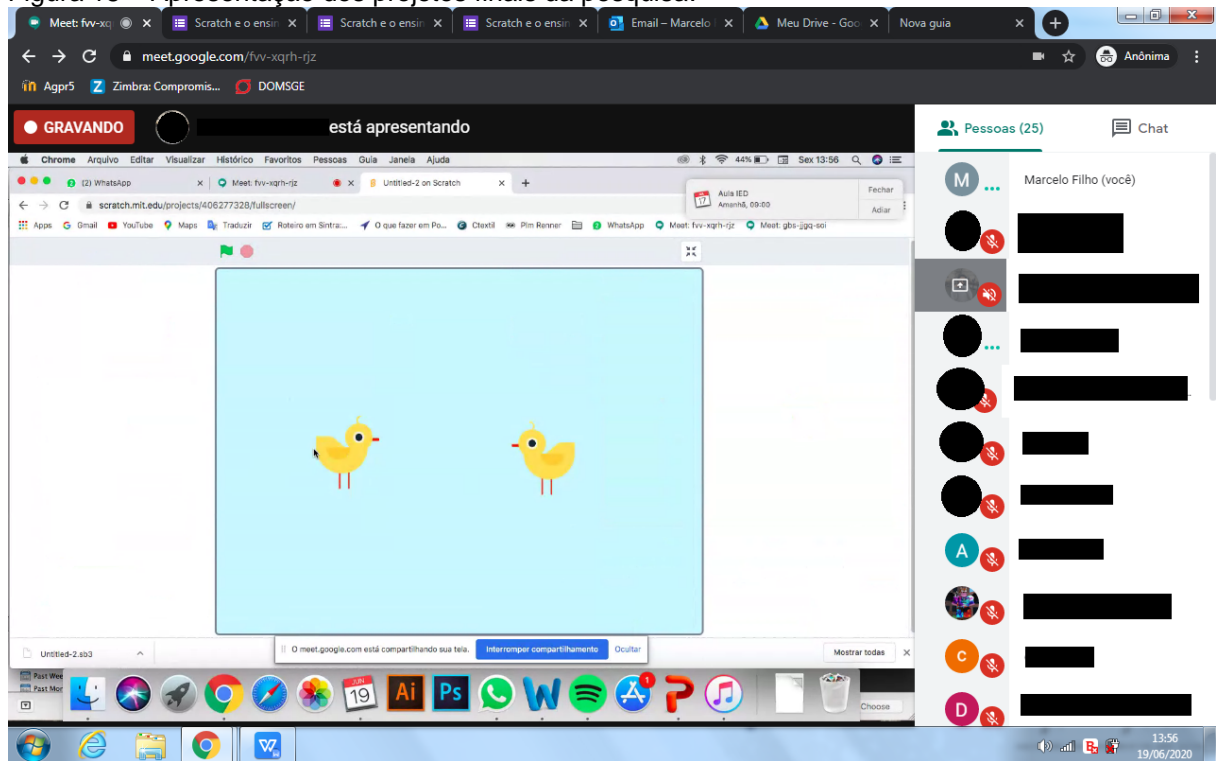


Fonte: Do autor.

A aula final (19/06) consistiu na apresentação dos projetos finais pelos estudantes, bem como a solicitação de preenchimento do questionário elaborado para avaliar sua participação no projeto. Devido à virtualidade das aulas, o questionário acabou sendo distribuído por meio da plataforma *Google Forms*, compartilhando o link de acesso com os estudantes. Após uma breve explicação de como responder às perguntas do formulário enviado, seguiu-se com as apresentações.

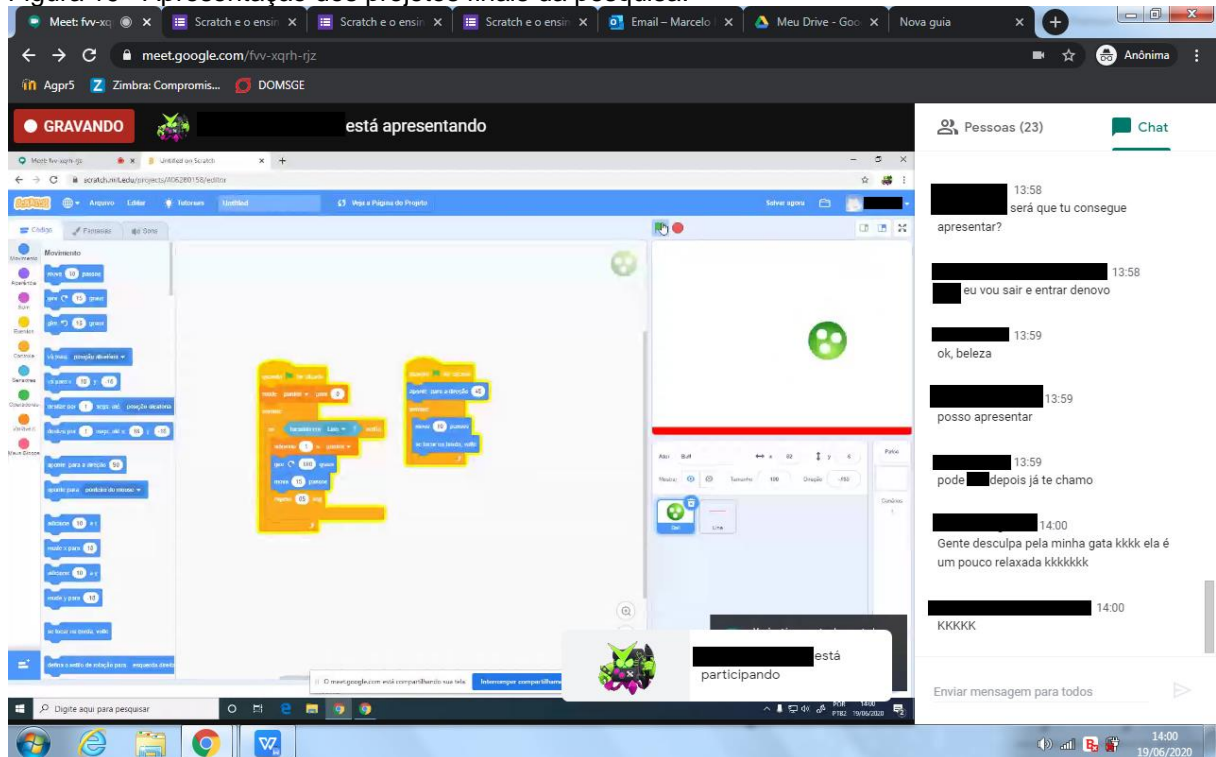
Os projetos, que seguiram uma temática de desenvolvimento livre, foram compartilhados com a turma por meio de apresentações individuais, como pode ser visto nas figuras 18 e 19.

Figura 18 – Apresentação dos projetos finais da pesquisa.



Fonte: Do autor.

Figura 19– Apresentação dos projetos finais da pesquisa.



Fonte: Do autor.

5.1.5 Análise dos resultados

Ao término da execução das aulas, foram analisados os dados recolhidos por meio da aplicação do questionário, realizando a tabulação e interpretação dos mesmos. Os resultados obtidos são apresentados detalhadamente no capítulo seguinte.

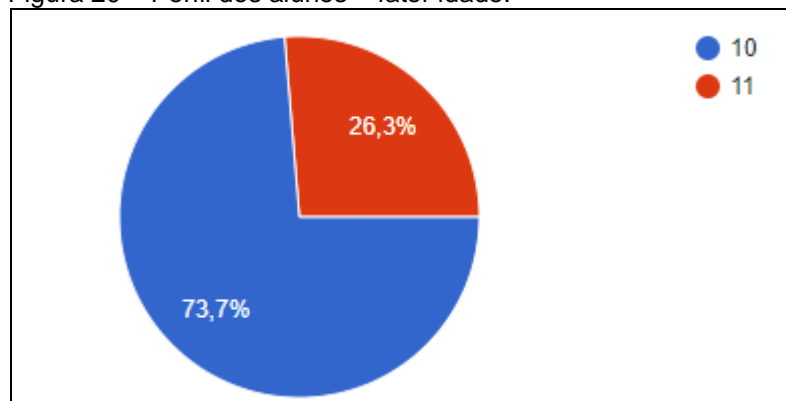
5.2 RESULTADOS OBTIDOS

O questionário desenvolvido na ferramenta *Google Forms* foi disponibilizado aos alunos participantes da última aula (19/06). Devido ao caráter assíncrono da execução das aulas, com a gravação das mesmas sendo realizadas para posterior compartilhamento, já que nem todos os estudantes conseguiram comparecer aos encontros, a amostragem final das respostas foi de um total de 21 alunos, similar a quantidade média de participantes por aula.

As duas primeiras questões foram referentes ao perfil dos alunos participantes, levando em consideração as variáveis idade e sexo.

A figura 20 demonstra que não houve uma variação significativa entre a idade dos estudantes, fator consequente da aplicação da pesquisa em uma única faixa de ensino (no caso, alunos do 5º ano do ensino fundamental). Houve a desconsideração de duas respostas devido ao preenchimento incorreto do formulário.

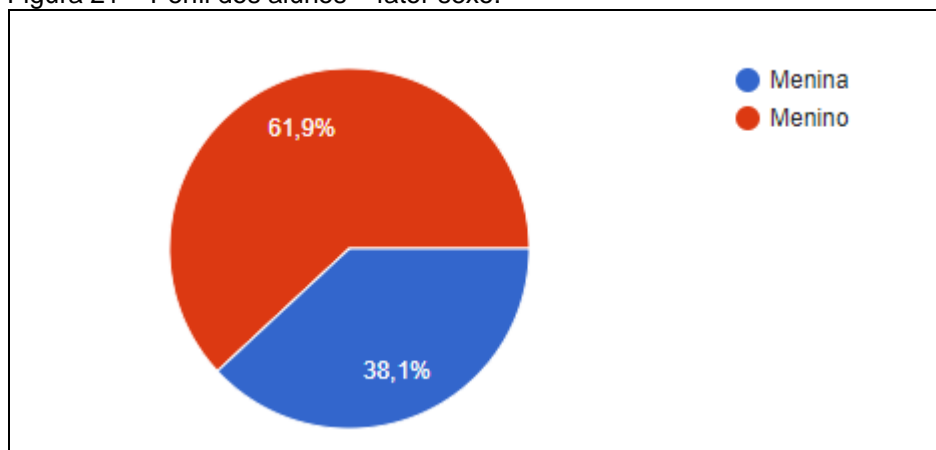
Figura 20 – Perfil dos alunos – fator idade.



Fonte: Do autor.

Com relação ao sexo dos participantes, observa-se que a amostragem possui uma margem superior de estudantes do sexo masculino (61,9%), com relação ao percentual de meninas participantes (38,1%), como pode ser visto na figura 21.

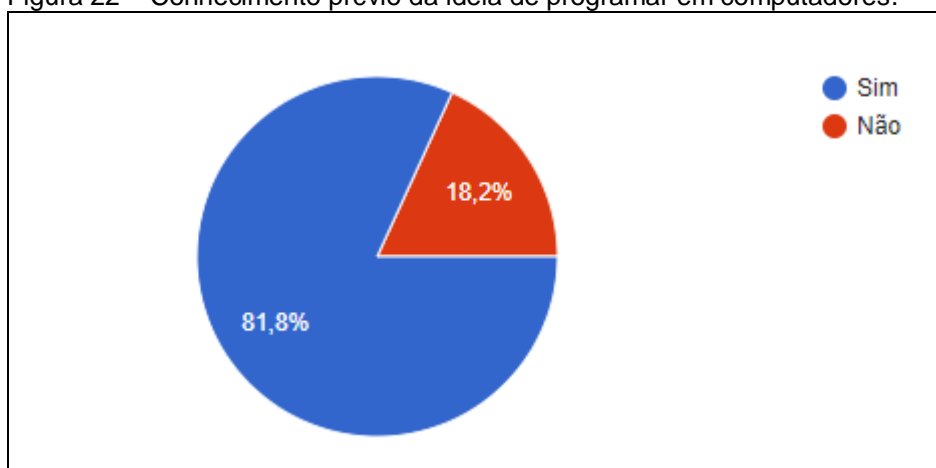
Figura 21 – Perfil dos alunos – fator sexo.



Fonte: Do autor.

A terceira questão pergunta aos participantes se eles já haviam ouvido falar em programação de computadores. O gráfico na figura 22 aponta que 57,1% das respostas enviadas continham um teor afirmativo quanto ao conhecimento prévio do tópico em questão.

Figura 22 – Conhecimento prévio da ideia de programar em computadores.

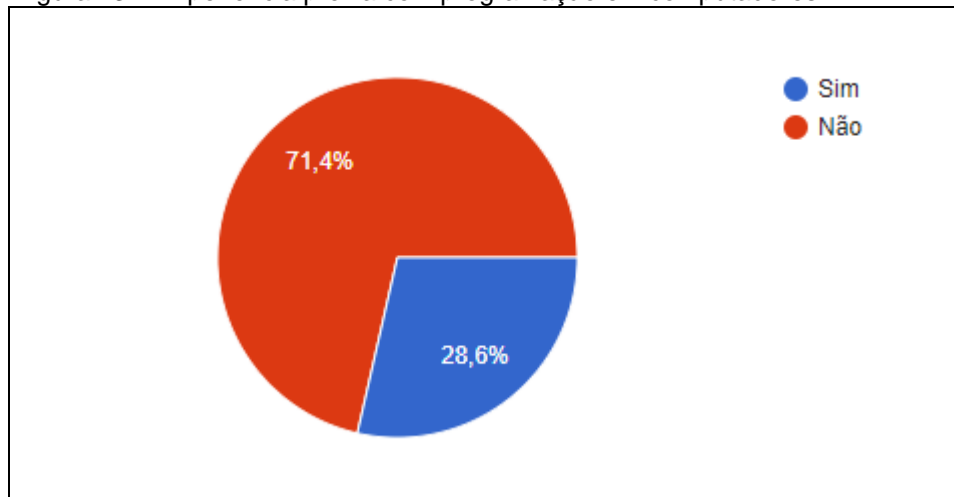


Fonte: Do autor.

Complementando a questão anterior, a quarta pergunta avalia a quantidade de estudantes que já possuíam alguma experiência com programação

antes da realização da pesquisa. Conforme visto na figura 23, nota-se que a maioria dos participantes (71,4%) não havia programado anteriormente.

Figura 23 – Experiência prévia com programação em computadores.

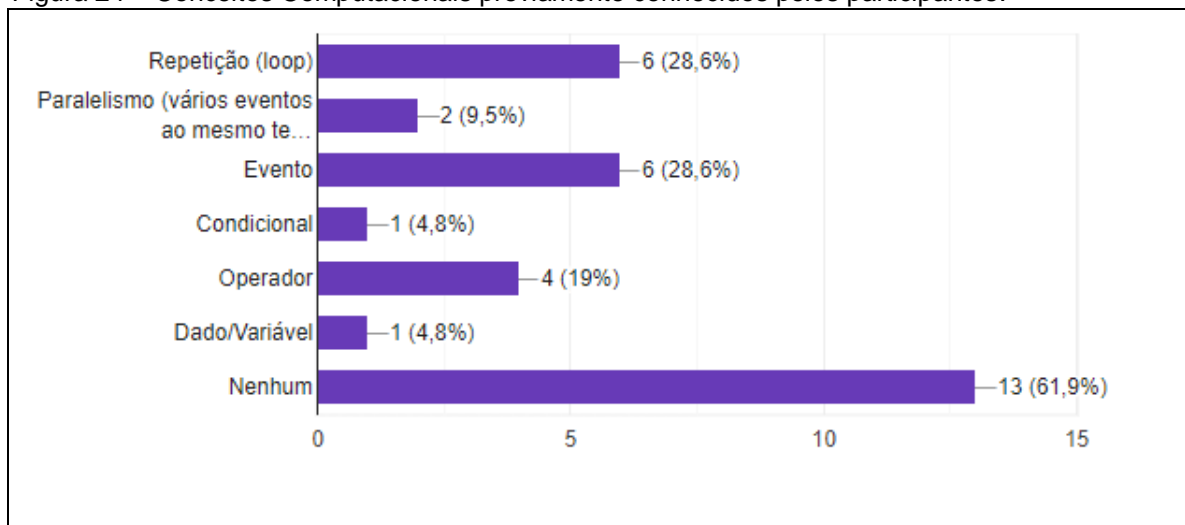


Fonte: Do autor.

A quinta questão solicitava aos participantes para informar quais conceitos relacionados ao pensamento computacional apresentados durante o projeto eles já conheciam, podendo selecionar uma ou mais das seguintes opções: “Repetição”, “Paralelismo”, “Evento”, “Condicional”, “Operador”, “Variável” ou “Nenhum”.

Nenhum dos conceitos atingiu uma parcela significativa da amostragem, com os mais conhecidos previamente sendo “Evento” e “Repetição” (seis alunos). A maioria dos estudantes (treze) respondeu não ter conhecimento de nenhum dos conceitos apresentados, conforme observado na figura 24.

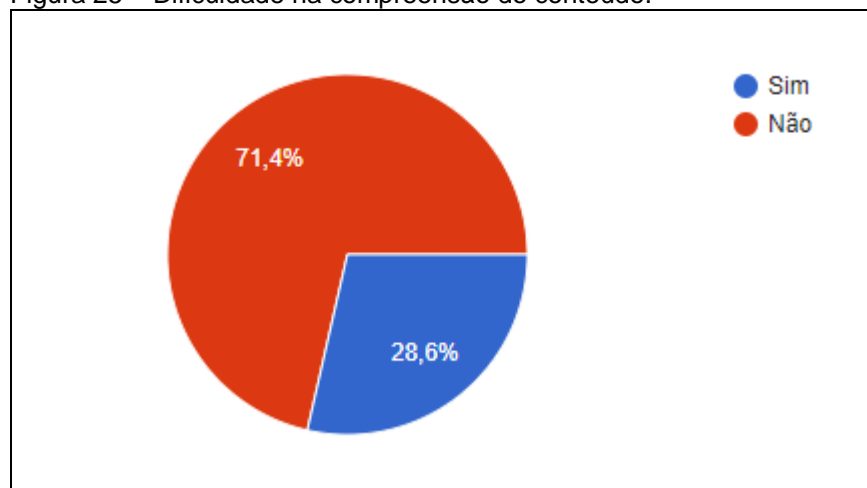
Figura 24 – Conceitos Computacionais previamente conhecidos pelos participantes.



Fonte: Do autor.

A sexta questão pedia aos alunos para informarem se houve alguma dificuldade em entender o conteúdo apresentado. A maioria (71,4%) não encontrou problemas que viessem a dificultar a compreensão do tema, como pode ser visto no gráfico da figura 25.

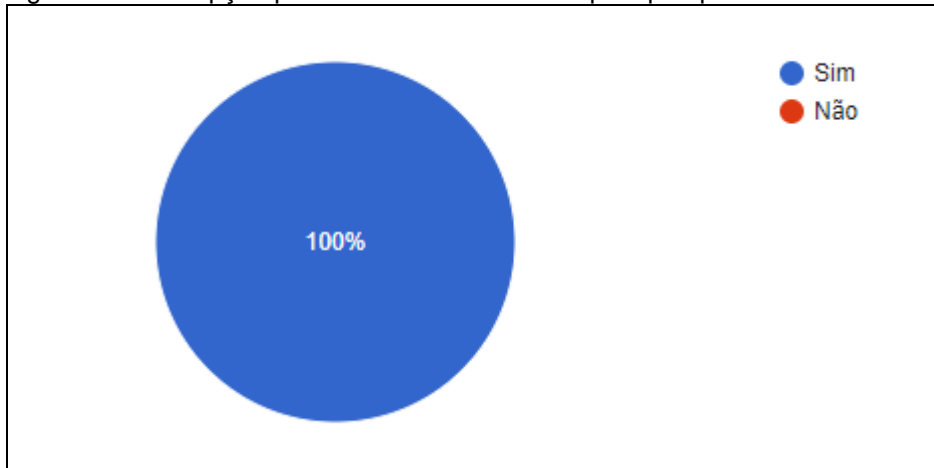
Figura 25 – Dificuldade na compreensão do conteúdo.



Fonte: Do autor.

A sétima questão dizia respeito ao modo com o qual o conteúdo foi apresentado. A totalidade dos alunos gostou da forma utilizada pela pesquisa, o que contribuiu para sua execução, visto que o engajamento dos participantes com relação ao projeto é potencializado mediante a aceitação positiva da abordagem utilizada (figura 26).

Figura 26 – Recepção positiva da forma utilizada pela pesquisa.

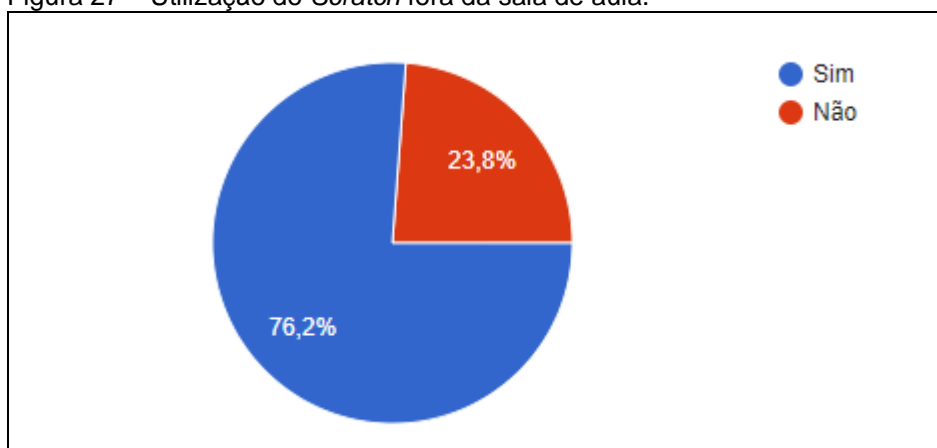


Fonte: Do autor.

A oitava questão solicitava aos estudantes para responderem, de modo discursivo, se, após a realização da pesquisa, haveria a vontade de continuar estudando sobre programação, justificando o porquê.

A maioria (dezesesseis participantes) confirmou que gostaria de continuar, afirmando que é um tema “legal” (seis ocorrências), gerando uma disposição para aprender mais (quatro ocorrências) ou que pretende desenvolver seus próprios jogos (duas ocorrências). Um dos participantes descreve que, apesar de ser algo de seu agrado, pode cansar devido à grande quantidade de informações visualizadas. Alguns dos participantes afirmam que não gostariam de continuar por não ser uma área de sua preferência ou por “perceber que não era para ele”.

Complementando a pergunta anterior, a nona questão diz respeito à ferramenta *Scratch*, perguntando aos participantes se utilizaram a plataforma além da sala de aula e/ou se utilizariam novamente após a realização do projeto. Dentre eles, 76,2% (dez alunos) afirmam que utilizaram e/ou continuariam utilizando a ferramenta além da sala de aula. Essa resposta está em sintonia com a quantidade de alunos que se interessaram em continuar estudando programação, como pode ser visto na figura 27.

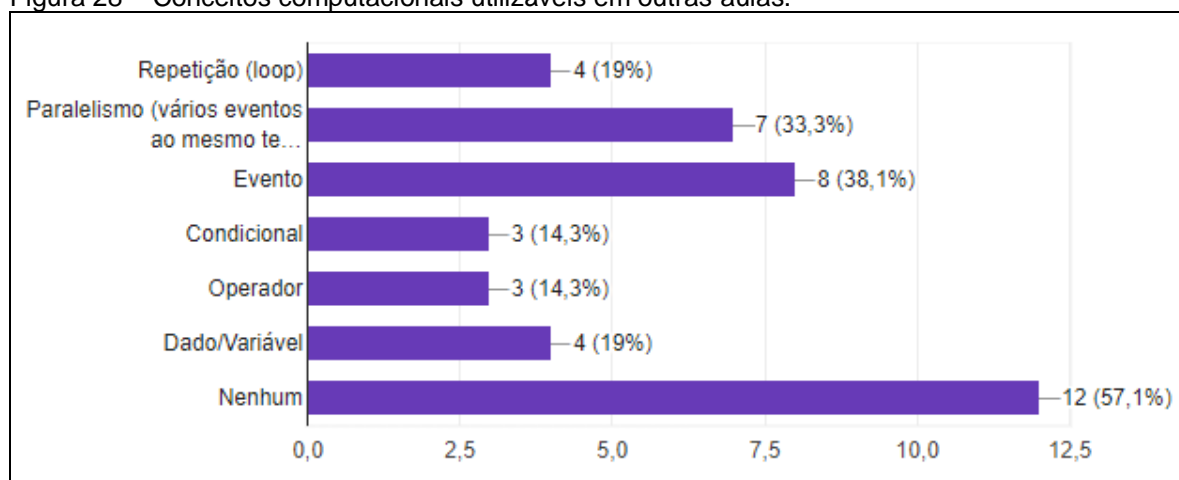
Figura 27 – Utilização do *Scratch* fora da sala de aula.

Fonte: Do autor.

A décima questão, discursiva, pedia aos estudantes para descrever seu interesse em participar de mais atividades envolvendo programação, sem, entretanto, determinar exclusivamente o uso do *Scratch*. Dezenove alunos responderam de forma afirmativa, pelo caráter “divertido” (quatro ocorrências), “legal” (três ocorrências), “importante” e “interessante” (uma ocorrência cada) do tópico, além de terem adquirido gosto pela plataforma (uma ocorrência) ou pela própria ideia de programar em si (uma ocorrência). Uma pessoa “não sabe” se participaria.

As duas questões seguintes utilizaram os mesmos conceitos definidos pela questão número cinco (“Repetição”, “Paralelismo”, “Evento”, “Condicional”, “Operador”, “Variável” ou “Nenhum”) para que os alunos respondessem quais dos tópicos vistos nas aulas poderiam ser utilizados em outras matérias. A figura 28 apresenta os resultados da décima primeira questão, observando a ênfase nos conceitos de “Evento” (oito pessoas) e “Paralelismo” (sete pessoas), embora doze pessoas não chegaram a associar possíveis utilizações dos conceitos em outras matérias.

Figura 28 – Conceitos computacionais utilizáveis em outras aulas.



Fonte: Do autor.

A outra questão vinculada aos tópicos, que solicitava justificar a questão anterior de forma discursiva, não era obrigatória, tendo um índice menor de preenchimento. Apesar de não definir claramente onde os conceitos poderiam ser aplicados, os estudantes compreendem a possibilidade de utilização dos conceitos na criação de conteúdo digital para o ensino de outros tópicos.

A última questão solicitava aos estudantes que descrevessem a importância do aprendizado dos conteúdos expostos em sala de aula. Nota-se que o aprendizado de programação aparenta ser algo que, em um contexto geral, é visto como importante, por parte dos alunos, sem, entretanto, aprofundar a justificativa.

5.3 ANÁLISE DOS RESULTADOS

Os resultados obtidos pela aplicação do questionário, juntamente com os projetos apresentados ao final das aulas, foram analisados e comparados com dados apresentados por pesquisas similares.

As pesquisas realizadas por Aono et al. (2017) e Meerbau-Salant et al. (2013) apontaram que a maioria dos alunos possui capacidade de compreender os conceitos de Ciência da Computação, estando em harmonia com os resultados obtidos nesta pesquisa, já que não houve dificuldades maiores na realização das atividades pelos estudantes. Entretanto, é necessário frisar que, apesar das considerações positivas com relação ao aprendizado dos conceitos, notou-se que

uma parcela maior dos participantes acabou não conseguindo associá-los com outras matérias, levando a considerar os resultados encontrados por Kalelioğlu e Gülbahar (2014), que afirmam que, dependendo do contexto empregado, a programação em *Scratch* não causa um impacto significativo nas habilidades de resolução de problemas. A não contextualização das atividades com tópicos de sala específicos acabou por não permitir a observação dos alunos em situações que pudessem empregar os conceitos de computação apresentados.

Dentre os conceitos que acabaram obtendo relevante compreensão pelos estudantes, destaca-se “Eventos” e “Paralelismo”. Isso se encontra de forma oposta ao resultado obtido por França e Amaral (2013), que, ao considerar o conceito de paralelismo como sendo não trivial de ser aprendido por alunos sem conhecimento prévio do tema, encontraram dificuldades em compreender tal conceito após a realização de uma oficina de três horas de duração. A conclusão estabelecida pelos pesquisadores foi a de que seria necessário um tempo maior dedicado à explicação do tema, fato que ficou constatado após a análise dos resultados desta pesquisa, com a disponibilidade do dobro de tempo (seis horas) mais o período entre aulas no qual os alunos poderiam continuar desenvolvendo suas atividades. Acredita-se que a abordagem utilizada pelo pesquisador durante a explanação do conceito, criando vários blocos de comando para execução simultânea dentro das atividades, também contribuiu para o seu aprendizado, ao empregar um foco maior de como a questão de paralelismo atua fortemente no desenvolvimento dos projetos.

Com o emprego dos recursos da programação visual, percebeu-se as vantagens elencadas por Saito et al. (2017), que afirma que tais recursos são benéficos aos iniciantes no mundo da programação, promovendo o estímulo necessário para que os aprendizes avancem conforme os conceitos são apresentados, sem a necessidade de prenderem-se à sintaxe da linguagem.

A utilização da ferramenta *Scratch* na realização das aulas como parte da metodologia da pesquisa mostrou-se positiva, com a maioria dos alunos não demonstrando dificuldades no entendimento do conteúdo, além da aceitação unânime dos participantes na forma de como o mesmo foi apresentado. Este fator está de acordo com o resultado apresentado por von Wangenheim et al. (2014) ao perceber que utilizar tal plataforma no ensino de programação, mesmo focando em uma faixa etária consideravelmente menor do que a dos participantes desta

pesquisa (6-7 anos de idade), proporciona um ambiente adequado e motivador aos participantes.

Além disso, ao propor um método visando uma aprendizagem ativa dos participantes, pode-se notar que houve uma motivação maior, por parte dos estudantes, em realizar as atividades. Embora o caráter não presencial das aulas impedisse a visualização do desenvolvimento das atividades dos alunos, percebeu-se que os mesmos sentiram-se animados em replicar o que era apresentado, visto que houve a criação em paralelo de pequenos projetos que faziam uso dos conceitos computacionais demonstrados, chegando inclusive a promover o compartilhamento com a turma.

O interesse em continuar utilizando o *Scratch* fora da sala de aula, tanto durante a realização da pesquisa, quanto após o término desta, além da disposição demonstrada nas respostas de continuar participando de atividades que possam envolver programação, denota similaridades com os resultados encontrados por Ribeiro (2017), onde a metodologia empregada, ao ter seu foco voltado para o caráter participativo dos estudantes, gerou um sentimento de coautoria com relação aos resultados produzidos.

Outro tópico a ser considerado foi a proposta de um projeto final com possibilidade de formato de desenvolvimento aberto como forma de avaliar o desempenho dos estudantes. Conforme destacado por Correa (2014), ao considerar o pensamento computacional um “conjunto de ferramentas para solucionar problemas complexos”, deve-se levar em consideração que a utilização de jogos (no caso desta pesquisa, juntamente com a construção de histórias) com a finalidade de levar ao desenvolvimento do pensamento computacional deve permitir ao participante planejar a solução de problemas, atitude observada pelos projetos desenvolvidos.

6 CONCLUSÃO

O pensamento computacional é uma ferramenta cognitiva com grande potencial de utilização no campo educacional, auxiliando no raciocínio lógico em resolução de problemas. Uma das formas de promover seu ensino consiste em apresentar conceitos de programação por intermédio da ferramenta *Scratch*. Neste projeto estudaram-se metodologias de ensino para planejar, aplicar e avaliar a utilização da ferramenta em sala de aula, realizando sua análise dentro do contexto de ensino dos conceitos de programação.

As dificuldades encontradas foram durante o processo de readaptação e execução de um novo cronograma devido aos imprevistos decorrentes da suspensão temporária das aulas, além de ser necessário viabilizar o uso de conferências digitais para a realização da pesquisa.

Apesar das dificuldades encontradas durante a realização da pesquisa, o objetivo principal, analisar a utilização da ferramenta *Scratch* em uma metodologia de ensino de pensamento computacional, foi alcançado após a realização das atividades finais por parte dos estudantes e a obtenção dos resultados por meio do questionário aplicado.

A pesquisa também foi direcionada conforme os objetivos específicos. O primeiro, descrever o conceito de pensamento computacional, foi concluído ao final do levantamento bibliográfico com a escrita da fundamentação teórica, posteriormente sendo apresentado em sala de aula para os estudantes. O segundo e o terceiro, que consistiam, respectivamente, em empregar a ferramenta *Scratch* como forma de desenvolver o aprendizado de raciocínio lógico e aplicar uma metodologia para avaliação do aprendizado da ferramenta empregada, foram concluídos ao término das aulas. O quarto objetivo, que dizia respeito a aplicar a programação em bloco ao processo de ensino de programação no laboratório de Informática do Colégio UNESC nos estudantes do quinto ano do Ensino Fundamental, acabou sendo parcialmente finalizado, visto que as aulas tiveram que mudar seu modelo de prática para os meios digitais. O último objetivo, verificar se a programação em bloco contribuiu para o aprendizado do pensamento computacional dos estudantes, analisando os testes e critérios empregados, foram concluídos após a aplicação do questionário ao final da execução do projeto.

Os resultados obtidos diziam respeito à aplicação da ferramenta para apresentar os conceitos do pensamento computacional, além de avaliar o aprendizado dos mesmos, tendo retorno positivo quanto à utilização da ferramenta e o seu objetivo, no caso, a demonstração de tópicos relacionados à programação, com os alunos conseguindo desenvolver as atividades durante a realização da pesquisa. Os resultados apresentados, tanto com relação à utilização de uma ferramenta visual estimulando o ensino de conceitos do pensamento computacional, quanto ao caso específico de aplicação da ferramenta *Scratch*, condizem com as demais pesquisas referentes ao assunto estudadas durante o levantamento bibliográfico. Outro ponto observado consistiu na metodologia aplicada, com o estudo de práticas construtivistas de atividades delegadas aos próprios estudantes contribuindo com o potencial de aprendizagem proposto pela ferramenta. Os pontos negativos elencados pela pesquisa consistem no imprevisto da utilização de plataformas digitais em detrimento das aulas presenciais, podendo ocasionar resultados diferentes daqueles esperados inicialmente, visto que o distanciamento entre o pesquisador e os alunos acaba excluindo o fator humano encontrado em aulas presenciais, bem como o acompanhamento visual dos estudantes.

Para dar continuidade aos estudos sugere-se como realização de trabalhos futuros:

- a) Realizar a pesquisa com *Scratch* por meio de aulas presenciais;
- b) Aplicar os conceitos do pensamento computacional em matérias do Colégio;
- c) Utilizar outra ferramenta como forma de comparar a preferência pelos participantes;
- d) Aplicar pesquisas similares com estudantes de faixas de idade diferentes.

REFERÊNCIAS

- ANDRADE, D.; CARVALHO, T.; SILVEIRA, J.; *et al.* **Proposta de Atividades para o Desenvolvimento do Pensamento Computacional no Ensino Fundamental.** Anais do Workshop de Informática na Escola, v. 1, n. 1, p. 169, 2013.
- AONO, A. H. et al. A Utilização do Scratch como Ferramenta no Ensino de Pensamento Computacional para Crianças. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO, 25., 2017, São José dos Campos. **Anais do XXV Workshop sobre Educação em Computação.** São Paulo: Sociedade Brasileira de Computação, 2017. p. 2169 - 2178.
- ARANTES, F. L.; AMIEL, Tel. **Nos rumos da autonomia tecnológica - desafios e lições aprendidas para a formação de jovens.** In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 3., WORKSHOP DE INFORMÁTICA NA ESCOLA, 20., 2014, Dourados. Anais... Porto Alegre: SBC, 2014. p. 308-317.
- ARAÚJO, A. L. S. O.; ANDRADE, W. L.; GUERRERO, D. D. S. (2015). "Pensamento Computacional sob a visão dos profissionais da computação: uma discussão sobre conceitos e habilidades". In: Anais dos Workshops do IV CBIE.
- BOCCONI, S. et al (Ed.). **Developing Computational Thinking in Compulsory Education:** Implications for policy and practice. Luxemburgo: Publications Office Of The European Union, 2016.
- BRACKMANN, C. P. **Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica.** Tese (Doutorado) – Programa de Pós-Graduação em Informática na Educação, Centro de Estudos Interdisciplinares em Novas Tecnologias na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017.
- BRASIL. Ministério da Educação. Secretaria da Educação Básica. Base nacional comum curricular. Brasília, DF, 2016. Disponível em: <<http://basenacionalcomum.mec.gov.br/#/site/inicio>>. Acesso em: nov. 2019.
- CAMPOS, F.R.; **Diálogo entre Paulo Freire e Seymour Papert:** A prática educativa e as tecnologias digitais de informação e comunicação. Tese (Doutorado em Letras), MACKENZIE, São Paulo, 2008.
- CIEB. Centro de Inovação para a Educação Brasileira, 2019. Disponível em: <<http://curriculo.cieb.net.br/>>. Acesso em: 1 dez. 2019.
- CORRÊA, E. F. S.. **Jogos eletrônicos e o desenvolvimento do pensamento computacional:** um estudo de caso. 2014. 79 f. Monografia (Graduação) - Curso de Ciência da Computação, Universidade de Brasília, Brasília, 2017.
- EASTERBROOK, S. (2014) **"From Computational Thinking to Systems Thinking: A conceptual toolkit for sustainability computing"**. Proceedings of the 2nd International Conference on Information and Communication Technologies for

Sustainability (ICT4S'2014), Stockholm, Sweden, 24-27 August, 2014.

FRANÇA, R. S. de; AMARAL, H. J. C. do. **Proposta Metodológica de Ensino e Avaliação para o Desenvolvimento do Pensamento Computacional com o Uso do Scratch**. Anais do Workshop de Informática na Escola, v. 1, n. 1, p. 179, 2013.

FRANÇA, R.; TEDESCO, P.; **Desafios e oportunidades ao ensino do pensamento computacional na educação básica no Brasil**. Anais dos Workshops do Congresso Brasileiro de Informática na Educação, v. 4, n. 1, p. 1464, 2015.

_____. **Pensamento computacional sob a perspectiva de licenciandos em computação**. Anais do Workshop de Informática na Escola, v. 23, n. 1, p. 795, 2017.

GOMES, Tancicleide C. S.; Melo, Jeane C.B. de Melo. **O Pensamento Computacional no Ensino Médio: Uma Abordagem Blended Learning**. 2016.

GOOGLE. **Google for Education: Blockly**. 2016. Página Inicial. Disponível em: <<https://developers.google.com/blockly/>>. Acesso em: 01 dez. 2019.

JONASSEN, D. O uso das tecnologias na Educação à Distância e as aprendizagem construtivista. Em aberto, Brasília, n.70, ano 16, abr./jun., 1996.

KALELIOĞLU, F.; GÜLBAHAR, Y. (2014). **The effects of teaching programming via scratch on problem solving skills: A discussion from learners' perspective**. *Informatics in Education*, 13(1), 33–50.

KELLEHER, C.; PAUSCH, R. 2005. **Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers**. ACM Computing Surveys (CSUR)

MALONEY J.; RESNICK M.; RUSK N.; SILVERMAN B.; EASTMOND E. **The Scratch Programming Language and Environment**. In ACM Transactions on Computing Education (TOCE), vol. 10, no. 4, November. 2010.

MEERBAUM-SALANT, Orni; Michal Armoni; MORDECAI Ben-Ari (2013). **Learning computer science concepts with scratch**. Computer Science Education v. 23.3: 239-264.

MENECH, Alexandre de. **TECNOLOGIAS NA EDUCAÇÃO: software livre no apoio ao processo de alfabetização e letramento dos alunos do 1º ao 3º ano do ensino fundamental do Colégio UNESC**. 2014. 88 f. TCC (Graduação) - Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, 2018.

MUCCIOLI, Cristina. O Comitê de Ética em Pesquisa (CEP) e as publicações científicas. **Arquivos Brasileiros de Oftalmologia**, [s.l.], v. 67, n. 2, p. 195-196, abr. 2004. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/s0004-27492004000200002>.

NATIONAL RESEARCH COUNCIL (U.S.) (ORG.). Report of a workshop of pedagogical aspects of computational thinking. Washington, D.C: National Academies Press, 2011.

NATIONAL RESEARCH COUNCIL (U.S.); COMMITTEE FOR THE WORKSHOPS ON COMPUTATIONAL THINKING. Report of a workshop on the scope and nature of computational thinking. Washington, D.C.: National Academies Press, 2010.

OLIVEIRA, M. L. S. DE; SOUZA, A. A. DE; BARBOSA, A. F.; BARREIROS, E. F. S. **Ensino de lógica de programação no ensino fundamental utilizando o Scratch:** um relato de experiência. Congresso da Sociedade Brasileira de Computação – CSBC, v. 12, p. 1–10, 2014.

PAPERT, S. *Mindstorms: Children, Computers, And Powerful Ideas*. Basic Books, 1980.

PARDAMEAN, B.; EVELIN, E., HONNI, H. (2011). **The effect of logo programming language for creativity and problem solving**. In Proceedings of the 10th WSEAS International Conference on E-Activities, E-ACTIVITIES'11, pages 151–156, Stevens Point, Wisconsin, USA.

PEREIRA, Amanda C.; FRANCO, Matheus E. **Desenvolvendo o pensamento computacional no ensino fundamental com Arduino e Scratch**. 2018.

RIBEIRO, Stephanie da Silva. **Um método para o desenvolvimento de software por crianças com o ambiente Scratch**. 2016. 97 f. TCC (Especialização) - Curso de Engenharia de Software, Universidade Federal do Pampa, Alegrete, 2016.

RUSK, N.; RESNICK, M.; MALONEY, J. "Learning with Scratch: 21 st Century Learning Skills". Lifelong Kindergarten Group. MIT Media Laboratory.

SAITO D.; WASHIZAKI H.; FUKAZAWA Y., "Comparison of Text-Based and Visual-Based Programming Input Methods for First-Time Learners", *JITE: Research*, vol. 16, pp. 209-226, 2017.

SAVI, Rafael. **Avaliação de jogos voltados para a disseminação do conhecimento**. 2011. 236 f. Tese (Doutorado) - Curso de Engenharia e Gestão do Conhecimento, Universidade Federal de Santa Catarina, Florianópolis, 2011.

SCRATCH, 2019. Disponível em: < <https://scratch.mit.edu/>>. Acesso em: dez. 2019.

SCRATCH BRASIL. (2014) "Sobre o Scratch – Você conhece o Scratch?" Site Scratch Brasil. Disponível em: < <http://www.scratchbrasil.net.br/index.php/sobre-o-scratch.html> >. Acesso em: nov. 2019.

SCRATCHED. **CREATIVE COMPUTING**: a design-based introduction to computational thinking. Cambridge: Harvard Graduate School Of Education, 2011.

SOCIEDADE BRASILEIRA DE COMPUTAÇÃO. Computação Itinerário Formativo. Disponível em:<<http://www.sbc.org.br/documentos-da-sbc/send/203-educacao-basica/1216-itinerario-formativo-da-computacao>>. Acesso em: 3 jun. 2019.

SOUZA, Raul Porto de. **USO DA BIBLIOTECA DE PROGRAMAÇÃO EM BLOCOS**

BLOCKY COMO FORMA DE AUXÍLIO AO APRENDIZADO DA DISCIPLINA DE ALGORITMOS E PROGRAMAÇÃO. 2018. 68 f. TCC (Graduação) - Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, 2018.

VON WANGENHEIM, Christiane Gresse; NUNES, Vinícius Rodrigues; SANTOS, Giovane Daniel dos. Ensino de Computação com SCRATCH no Ensino Fundamental – Um Estudo de Caso. **Revista Brasileira de Informática na Educação**, [s.l.], v. 22, n. 03, p.115-125, 23 nov. 2014. Sociedade Brasileira de Computacao - SB. <http://dx.doi.org/10.5753/rbie.2014.22.03.115>.

WEINTROP, D.; WILENSKY, U. (2017) **Comparing Blocks-based and Text-based Programming in High School Computer Science Classrooms.** Transactions on Computing Education, 18(1).

_____. To Block or not to Block, That is the Question: Students' Perceptions of Blocks-based Programming. In: INTERNATIONAL CONFERENCE ON INTERACTION DESIGN AND CHILDREN, 14., 2015, Boston. **Proceedings of IDC 2015: The 14th International Conference on Interaction Design and Children.** Boston: Association For Computing Machinery, Inc, 2015. p. 199 - 208.

WING, J. M.; Computational Thinking. 2006. Disponível em: <cs.cmu.edu/~15110-s13/Wing06-ct.pdf>. Acesso em 30.05.2019.

_____. Computational Thinking: What and Why? , 17. out. 2010. Disponível em: <<http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>>. Acesso em: 1 dez. 2019.

_____. Computational thinking and thinking about computation. 2008. Disponível em: < <https://www.cs.cmu.edu/~wing/publications/Wing08a.pdf>>. Acesso em 1 dez. 2019.

APÊNDICES

APÊNDICE A – Questionário aplicado para avaliação da pesquisa

Scratch e o ensino de Programação

Qual sua idade?

Você é...

- Menina
- Menino

Você já tinha ouvido falar em programação de computadores antes?

- Sim
- Não

Você já teve alguma experiência com programação antes?

- Sim
- Não

Você conhecia algum dos conceitos apresentados? Fique à vontade para selecionar mais de um item

- Repetição (loop)
- Paralelismo (vários eventos ao mesmo tempo)
- Evento
- Condicional
- Operador
- Dado/Variável
- Nenhum

Você teve dificuldade em entender o conteúdo apresentado?

- Sim
- Não

Você gostou da forma como o conteúdo foi apresentado?

- Sim
- Não

Depois dessa pesquisa, você gostaria de continuar estudando sobre programação (Sim ou Não)? Por quê? **(Discursiva)**

Você usou/continuará utilizando o Scratch além da sala de aula?

- Sim
- Não

Você participaria de mais atividades envolvendo programação como essa? Por quê? **(Discursiva)**

Você acha que o que foi visto nas aulas pode ser utilizado em outras matérias? Fique à vontade para selecionar mais de um item

- Repetição (loop)
- Paralelismo (vários eventos ao mesmo tempo)
- Evento
- Condicional
- Operador
- Dado/Variável
- Nenhum

Caso tenha selecionado algum item na resposta anterior, por quê? **(Discursiva)**

Por que o conteúdo apresentado nas aulas é importante para ser aprendido? **(Discursiva)**

APÊNDICE B – Artigo

Scratch e o ensino do pensamento computacional aplicados aos alunos do Colégio Unesc

Luciano Antunes¹, Marcelo Milioi Bristot Filho¹

¹Curso de Ciência da Computação – Universidade do Extremo Sul Catarinense (UNESC) – Criciúma – SC – Brasil

luc@unesc.br, marcelombfilho@outlook.com

Abstract. *Computational thinking potential as a cognitive tool can be applied in the children's teaching as a logical reasoning development assistant. Understanding Computational concepts helps the problem solving in others branches of knowledge. As a way to analyse this influence, programming teaching with a block-oriented visual approach, applying those concepts without contextual specificities, can contribute to this thinking method learning process. This research evaluates the utilization of Scratch within a teaching methodology intended to the computational concepts learning.*

Resumo. *O potencial do pensamento computacional enquanto ferramenta cognitiva pode ser aplicado no ensino de crianças como forma de auxiliar no desenvolvimento do raciocínio lógico. A compreensão de conceitos extraídos da Computação ajuda na resolução de problemas de outras áreas do conhecimento. Como forma de analisar essa influência, o ensino de programação com uma abordagem visual orientada por blocos, ao aplicar tais conceitos removendo o contexto sintático, pode contribuir no processo de aprendizagem desse método de pensamento. Esta pesquisa avalia a utilização da ferramenta Scratch, integrada em uma metodologia de ensino voltada para a aprendizagem de conceitos computacionais.*

1. Introdução

A relação dos seres humanos com a Computação permitiu o desenvolvimento de metodologias e ferramentas que moldaram a realidade na qual estamos inseridos [SBC 2018]. Conceitos e fundamentos referentes a essa área de estudo formam a base de uma habilidade conhecida como “pensamento computacional”, utilizada na resolução de problemas e desenvolvimento de sistemas, dentro de qualquer área de conhecimento [Wing 2006]. Com isso, pode-se promover tal habilidade como forma de contribuir para o desenvolvimento do raciocínio lógico de indivíduos, notoriamente em estudantes em processo de formação do pensamento crítico.

Conforme denotado por Arantes et al. (2014), há a necessidade de formar os jovens rumo a uma “autonomia tecnológica”, para que possam compreender o funcionamento de novas tecnologias. França et al. (2014) aponta que diversas ações já foram realizadas como forma de estimular o ensino do pensamento computacional, seja na inclusão direta na grade curricular, como também em alternativas que aproveitem da interdisciplinaridade da Computação. Mesmo sem existir consenso no melhor método

para promover tal ensino, nota-se a existência de três pilares básicos que norteiam tais atividades, a saber: abstração, automação e análise [Andrade et al., 2013].

Uma das formas encontradas para o ensino desses conceitos consiste em utilizar ferramentas que possam situar o usuário no contexto da programação, já que esta consiste em uma aplicação prática dos fundamentos do pensamento computacional. Como forma de remover empecilhos que possam dificultar a aprendizagem, como elementos de sintaxe, por exemplo, pode-se empregar ferramentas com abordagens visuais, que codificam elementos gramaticais em blocos de ações interativos para o usuário, criando um ambiente de fácil utilização [Weintrop e Wilensky, 2015].

A ferramenta *Scratch*, ao empregar este paradigma, permite realizar a execução de aulas com a exposição dos conceitos vinculados com o pensamento computacional. Sua utilização, conforme destaca Pereira et al. (2012), auxilia na melhor compreensão de conceitos de programação de uma forma simples e divertida, permitindo o exercício da criatividade e do raciocínio lógico e, por consequência, preparando-os na resolução de problemas computacionais propostos.

Baseando-se na ideia de propor o ensino do pensamento computacional para jovens, propôs-se realizar a análise da aplicação de uma metodologia que utilize a plataforma *Scratch* para esse fim, tendo como amostragem estudantes do 5º ano do Ensino Fundamental.

2. Fundamentação Teórica

O pensamento computacional ganhou notoriedade em meados da primeira década de 2000, quando Jeanette Wing, por meio de um artigo publicado em 2006, intitulado “Computational Thinking”, assinalou-o como uma habilidade fundamental para todas as pessoas, pela possibilidade de resolução de problemas. Apesar de não existir um consenso na definição precisa do termo, pode-se entendê-lo como uma “ferramenta cognitiva”, com os denominadores em comum na definição do termo incluindo a ideia de um “processo de pensamento”, com conceitos como abstração e decomposição.

A Sociedade Brasileira de Computação (SBC), Sociedade Científica com finalidade de incentivar o progresso da área de Computação no Brasil, descreve o pensamento computacional como a “habilidade de compreender, definir, modelar, comparar, solucionar, automatizar e analisar problemas (e soluções) de forma metódica e sistemática” [SBC 2020].

Reconhecendo a importância de incluir o ensino de conceitos vinculados com a Computação no currículo da Educação Básica, a SBC elaborou um documento apresentando uma proposta de competências e habilidades a serem ensinadas nas escolas, contribuindo no desenvolvimento da análise crítica e capacidade de solução de problemas [SBC 2017]. Tendo como uma das bases esse documento, o Centro de Inovação para a Educação Brasileira (CIEB) desenvolveu um currículo referencial com as diretrizes e orientações objetivando a construção de habilidades referentes aos temas de Tecnologia e Computação, tendo como um dos eixos o pensamento computacional, que, por sua vez, dividia-se em quatro habilidades: Abstração, Algoritmos, Decomposição e Reconhecimento de Padrões (CIEB 2019).

Como forma de promover o pensamento computacional, pode-se realizar o ensino de programação, por consistir em uma aplicação prática de seus fundamentos. Um dos formatos desse ensino consiste na utilização de ferramentas visuais, que acabam sendo mais adequada para iniciantes não familiarizados com o mundo da programação [Washizaki e Fukuzawa 2017]. Linguagens de programação orientadas por blocos consistem em variantes da programação visual, onde os ambientes promovem a utilização de peças representando funções, encaixando entre si para formar os componentes do código, com o usuário desenvolvendo seus programas enquanto recebendo o retorno visual [Weintrop e Wilensky 2015].

A ferramenta *Scratch*, um ambiente web gratuito de programação visual, aplica os conceitos da programação visual na criação de *scripts*, programas criados com a formação de blocos de comando, com uma interface gerando o retorno imediato às ações do usuário, além de ignorar questões de sintaxe [Meerbaum-Salant et al. 2014].

Os conceitos nos quais se baseia a ferramenta *Scratch* fazem parte de uma abordagem conhecida como metodologia construtivista. Essa metodologia sugere que o aluno em questão deve promover a construção de seu conhecimento de forma ativa, e, dentro do contexto do ensino de programação, estabelece uma relação direta de interação entre o estudante e a ferramenta de ensino [Gomes 2002]. O método construtivista em questão torna-se evidente na plataforma, ao estimular os usuários a criarem seus próprios projetos de forma interativa [Castro e Koscianski 2017].

3. Trabalhos Correlatos

A utilização do *Scratch* como forma de promover o ensino de conceitos de Computação foi abordada em diversas propostas. Aono et al. (2017) aplicou a plataforma, visando o ensino do pensamento computacional para estudantes do ensino fundamental. Visando alcançar tal objetivo, empregou-se uma metodologia expositiva aliada à utilização da ferramenta, consistindo em explicações teóricas dos conceitos seguindo de atividades práticas relacionadas aos tópicos recém-apresentados. Como agente motivador, propôs-se a criação de um jogo, desenvolvido progressivamente conforme os tópicos eram apresentados e as atividades realizadas em sala de aula.

Outra método aplicando a ferramenta foi proposto na pesquisa de von Wangenheim et al. (2014), consistindo no desenvolvimento de uma unidade instrucional (lições, exercícios e atividades em torno de um tema logicamente sequenciadas) para promover o ensino de computação nos anos iniciais da educação básica. Estabeleceu-se a criação de uma versão interativa de “Chapeuzinho Vermelho”, fazendo uso da interdisciplinaridade como definidora do conteúdo das aulas de computação.

Oliveira et al. (2014) utilizaram o *Scratch* como auxiliar na compreensão de algoritmos, possibilitando um aprendizado mais intuitivo e visualmente agradável, estimulando a criatividade e imaginação dos participantes. Com o formato de um curso ministrado para estudantes do ensino fundamental, o conteúdo de lógica de programação foi ministrado com a possibilidade de criação de jogos e animações na ferramenta, estimulando os alunos a desenvolverem conforme seus interesses. Outra proposta de ensino com a metodologia aplicada em cursos foi descrita por Meerbaum-Salant et al. (2013), por estudantes dentro da faixa etária de 14-15 anos. Separados em

duas classes, houve a realização de duas horas de aula por semana durante um semestre, com o desenvolvimento de materiais de apoio baseados no *Scratch*, alcançando resultados positivos no que tange ao aprendizado de conceitos computacionais pelos estudantes.

Dentro deste projeto, propôs-se o ensino do pensamento computacional por meio de uma metodologia expositiva com posterior aplicação prática dos conceitos apresentados, de forma a explorar as potencialidades da ferramenta no desenvolvimento de atividades. Como forma de incentivar a produção dos estudantes, definiu-se que haveria o desenvolvimento de um projeto final para apresentação baseando-se nas atividades realizadas durante as aulas da pesquisa.

4. Estudo de Caso Realizado

A realização desse projeto consistiu na aplicação de aulas expondo conceitos do pensamento computacional para estudantes do 5º ano do Ensino Fundamental do Colégio UNESC, por meio da utilização da ferramenta *Scratch*, para posteriormente realizar a análise da metodologia empregada.

Para que o projeto fosse realizado, inicialmente realizou-se uma parceria com a direção do Colégio UNESC para que fosse cedido um dos laboratórios de informática para execução das aulas, possibilitando, inclusive, integrá-las à grade curricular da turma do 5º ano, composta por alunos na faixa de idade entre 10-11 anos.

Devido à participação de crianças na pesquisa, foi necessária a submissão do projeto ao Comitê de Ética em Pesquisa (CEP) da UNESC, como forma de manter os aspectos éticos da pesquisa. Após a aprovação por parte do Comitê, seguiu-se com o planejamento das aulas.

4.1. Planejamento das aulas

Visando alinhar o projeto com a forma com que as aulas são realizadas, optou-se por manter o padrão de utilização da sala com uma abordagem expositiva dos conteúdos, seguida por realização de atividades para aplicação prática dos tópicos apresentados. O tempo disponibilizado para a execução do projeto, uma hora por semana, durante seis semanas, contribuiu para uma melhor distribuição de conceitos apresentados durante o planejamento.

O conteúdo das aulas foi baseado nos conceitos que norteiam a plataforma *Scratch*. Tais conceitos são perceptíveis na forma como a plataforma divide seus blocos, separados por categorias que empregam funções para a realização dos programas. Essas categorias são: sequência, repetições, paralelismo, eventos, condicionais, operadores e dados. Complementando os conceitos elencados a partir dessas categorias, procurou-se nortear a exposição dos conceitos de acordo com o guia “*Creative Computing – a design-based introduction to computational thinking*”, uma publicação escrita pelo grupo *ScratchEd* (2011) como forma de prover um currículo para auxiliar professores no desenvolvimento de conteúdo para exposição do pensamento computacional com base na ferramenta *Scratch*.

Para determinar a forma de aplicação da ferramenta *Scratch*, levou-se em consideração a abordagem construtivista com a qual ela foi idealizada. Ao colocar a

criança como sujeito ativo na construção do conhecimento, entende-se que o desenvolvimento de projetos individuais por parte dos estudantes contribui para promover a criatividade dos participantes, estimulando de forma positiva na aprendizagem dos conceitos no decorrer dos encontros.

Como forma de avaliar o desempenho final dos participantes, assim como para observar a eficácia da pesquisa pela perspectiva dos próprios estudantes, definiu-se que haveria a aplicação de um questionário ao final do projeto. Esse questionário baseou-se nas perguntas desenvolvidas por Pereira e Franco (2018), ao levarem em consideração os pontos elencados por Savi (2011) para modelar a avaliação da aprendizagem de uma metodologia de ensino de pensamento computacional.

Com isso, ficou definido como seria dividida a realização das aulas, conforme pode ser visto na Tabela 1.

Tabela 1. Cronograma proposto para realização do projeto

Aula	Conteúdo
1ª	Introdução à ferramenta, apresentação e descrição do projeto
2ª	Apresentação de conteúdo por meio de tutoriais da plataforma, com posterior replicação por parte dos estudantes
3ª	Continuação da exibição de conteúdo por meio de tutoriais da plataforma, com posterior replicação por parte dos estudantes
4ª	Realização dos projetos por parte dos estudantes
5ª	Continuação do desenvolvimento dos projetos por parte dos estudantes
6ª	Apresentação dos projetos finais e preenchimento dos formulários

4.2. Execução do projeto

A execução das aulas, prevista inicialmente para ocorrer de forma presencial durante os meses de abril e maio, acabou sendo adiada devido às medidas restritivas adotadas pelo colégio em virtude do surto de coronavírus, com o cancelamento imediato das aulas presenciais.

A retomada das aulas ocorreu de forma virtual, por meio da plataforma *Google Meet* para a realização dos encontros. Com isso, decidiu-se que as aulas seriam executadas às sextas, com a primeira aula marcada para 08/05. Apesar dos contratempos, o planejamento inicial para a exposição dos conceitos pode ser aplicado.

A realização das atividades foi realizada por meio da ferramenta de apresentação de tela da plataforma de reuniões virtuais (Figura 1), com os estudantes podendo compartilhar suas criações para os colegas conforme o desenvolvimento era realizado.

O último encontro (inicialmente previsto para 12/06, porém realizado em 19/06) consistiu na apresentação dos projetos finais desenvolvidos nas duas aulas anteriores pelos estudantes, permitindo observar de forma mais visível o resultado da aplicação das aulas.

Tabela 2. Respostas relacionadas à pesquisa obtidas pelo questionário.

Questão	Sim	Não
Conhecimento prévio da ideia de programar em computadores	81.8%	18.2%
Experiência prévia com programação em computadores	71.4%	28.6%
Dificuldade na compreensão do conteúdo	71.4%	28.6%
Recepção positiva da forma utilizada pela pesquisa	100%	0%
Utilização do <i>Scratch</i> fora da sala de aula	76.2%	23.8%

Questões descritivas também foram empregadas, visando extrair dados de caráter subjetivo com a finalidade de compreender o ponto de vista dos participantes. Ao serem questionados sobre a vontade de continuar estudando sobre programação, a maioria dos estudantes afirmou que sim, gostaria de dar continuidade, descrevendo o tema como “legal”, gerando uma disposição para aprender mais. Outra questão abordada, complementando o tópico, solicitava aos participantes para que descrevessem o interesse em participar de mais atividades envolvendo programação, sem, entretanto, determinar exclusivamente o uso do *Scratch*. A maioria respondeu de forma afirmativa, tanto pelo caráter “divertido”, pela importância da área.

Com relação ao conteúdo apresentado, realizou-se uma análise entre os conceitos previamente conhecidos pelos participantes, e quais poderiam ser aplicados em outras aulas, procurando compreender como os estudantes observariam a interdisciplinaridade dos conceitos computacionais. Os resultados obtidos referentes podem ser observados na tabela 3.

Tabela 3. Respostas referentes aos conceitos obtidas pelo questionário.

Conceitos computacionais	Conhecidos	Aplicáveis
Repetição (<i>loop</i>)	6 (28.6%)	4 (19%)
Paralelismo (vários eventos ao mesmo tempo)	2 (9.5%)	7 (33.3%)
Evento	6 (28.6%)	8 (38.1%)
Condicional	1 (4.8%)	3 (14.3%)
Operador	4 (19%)	3 (14.3%)
Dado/Variável	1 (4.8%)	4 (19%)
Nenhum	13 (61.9%)	12 (57.1%)

Após a tabulação dos dados obtidos, pode-se realizar sua análise, juntamente com os trabalhos apresentados pelos estudantes, e compará-los com resultados apresentados por pesquisas similares.

A compreensão dos conceitos de Ciência da Computação observado na maioria dos alunos está em harmonia com os resultados obtidos pelas pesquisas realizadas por Aono et al. (2017) e Meerbaum-Salant et al. (2013), sem maiores dificuldades na realização das atividades pelos estudantes. Entretanto, mesmo com as considerações positivas acerca do aprendizado dos conceitos, notou-se que uma parcela maior dos

participantes acabou por não conseguir associá-los com outras matérias, levando a considerar os resultados encontrados por Kalelioğlu e Gülbahar (2014), ao afirmarem que, dependendo do contexto empregado, a programação com a plataforma *Scratch* não causa um impacto significativo na resolução de problemas.

O emprego de recursos da programação visual permitiu reconhecer as vantagens elencadas por Saito et al. (2017), que afirma que tais recursos beneficiam os iniciantes no mundo da programação, promovendo o estímulo necessário aos aprendizes para que avancem conforme os conceitos são apresentados, sem prenderem-se a questões sintáticas da linguagem, por exemplo.

Utilizar a ferramenta *Scratch* como parte da metodologia de pesquisa mostrou-se positivo, sem a demonstração de dificuldades no entendimento do conteúdo pela maioria dos alunos, além da aceitação unânime dos participantes com relação à forma de como o mesmo foi apresentado, consistindo em um fator que encontra-se de acordo com o resultado apresentado por von Wangenheim et al. (2014), ao perceber que tal plataforma no ensino de programação proporciona um ambiente adequado e motivador aos participantes. Além disso, notou-se que o interesse na continuação do uso da plataforma fora da sala de aula, tanto durante a realização da pesquisa quanto após seu término, denota similaridades com os resultados vistos em Ribeiro (2017), onde a metodologia empregada, ao ter seu foco voltado para o caráter participativo dos estudantes, criou a sensação de coautoria na produção dos resultados.

6. Conclusão

Este trabalho realizou a análise da aplicação de uma metodologia de ensino do pensamento computacional por meio da utilização da ferramenta *Scratch* aplicada em estudantes do 5º ano do Colégio UNESC, visando observar como a programação em blocos, baseando-se em um processo construtivista de aquisição de conhecimento, poderia contribuir no estímulo do raciocínio lógico.

Apesar das dificuldades encontradas para a realização do projeto, levando em consideração modificações na forma de execução, pode-se perceber com os resultados que houve uma aceitação positiva por parte dos alunos com relação à metodologia de execução, bem como com a ferramenta utilizada, levando à compreensão dos conceitos apresentados. Desta forma, concluiu-se que a aplicação da pesquisa contribuiu para o aprendizado de conceitos referentes à programação e blocos, e, por conseguinte, ao estímulo do pensamento computacional.

Referências

- ANDRADE, D.; CARVALHO, T.; SILVEIRA, J.; *et al.* Proposta de Atividades para o Desenvolvimento do Pensamento Computacional no Ensino Fundamental. Anais do Workshop de Informática na Escola, v. 1, n. 1, p. 169, 2013.
- AONO, A. H. et al. A Utilização do Scratch como Ferramenta no Ensino de Pensamento Computacional para Crianças. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO, 25., 2017, São José dos Campos. Anais do XXV Workshop sobre Educação em Computação. São Paulo: Sociedade Brasileira de Computação, 2017. p. 2169 - 2178.

- ARANTES, F. L.; AMIEL, Tel. Nos rumos da autonomia tecnológica - desafios e lições aprendidas para a formação de jovens. In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 3., WORKSHOP DE INFORMÁTICA NA ESCOLA, 20., 2014, Dourados. Anais... Porto Alegre: SBC, 2014. p. 308-317.
- BRASIL. Ministério da Educação. Secretaria da Educação Básica. Base nacional comum curricular. Brasília, DF, 2016. Disponível em: <<http://basenacionalcomum.mec.gov.br/#/site/inicio>>. Acesso em: nov. 2019.
- CIEB. Centro de Inovação para a Educação Brasileira, 2019. Disponível em: <<http://curriculo.cieb.net.br/>>. Acesso em: 1 dez. 2019.
- CORRÊA, E. F. S.. Jogos eletrônicos e o desenvolvimento do pensamento computacional: um estudo de caso. 2014. 79 f. Monografia (Graduação) - Curso de Ciência da Computação, Universidade de Brasília, Brasília, 2017.
- FRANÇA, R. S. de; AMARAL, H. J. C. do. Proposta Metodológica de Ensino e Avaliação para o Desenvolvimento do Pensamento Computacional com o Uso do Scratch. Anais do Workshop de Informática na Escola, v. 1, n. 1, p. 179, 2013.
- FRANÇA, R.; TEDESCO, P.; Desafios e oportunidades ao ensino do pensamento computacional na educação básica no Brasil. Anais dos Workshops do Congresso Brasileiro de Informática na Educação, v. 4, n. 1, p. 1464, 2015.
- GOMES, Tancicleide C. S.; Melo, Jeane C.B. de Melo. O Pensamento Computacional no Ensino Médio: Uma Abordagem Blended Learning. 2016.
- KALELIOĞLU, F.,; GÜLBAHAR, Y. (2014). The effects of teaching programming via scratch on problem solving skills: A discussion from learners' perspective. *Informatics in Education*, 13(1), 33–50.
- MEERBAUM-SALANT, Orni; Michal Armoni; MORDECAI Ben-Ari (2013). Learning computer science concepts with scratch. *Computer Science Education* v. 23.3: 239-264.
- OLIVEIRA, M. L. S. DE; SOUZA, A. A. DE; BARBOSA, A. F.; BARREIROS, E. F. S. Ensino de lógica de programação no ensino fundamental utilizando o Scratch: um relato de experiência. Congresso da Sociedade Brasileira de Computação – CSBC, v. 12, p. 1–10, 2014.
- PEREIRA, Amanda C.; FRANCO, Matheus E. Desenvolvendo o pensamento computacional no ensino fundamental com Arduino e Scratch. 2018.
- RIBEIRO, Stephanie da Silva. Um método para o desenvolvimento de software por crianças com o ambiente Scratch. 2016. 97 f. TCC (Especialização) - Curso de Engenharia de Software, Universidade Federal do Pampa, Alegrete, 2016.
- SAITO D.; WASHIZAKI H.; FUKAZAWA Y., "Comparison of Text-Based and Visual-Based Programming Input Methods for First-Time Learners", *JITE: Research*, vol. 16, pp. 209-226, 2017.
- SAVI, Rafael. Avaliação de jogos voltados para a disseminação do conhecimento. 2011. 236 f. Tese (Doutorado) - Curso de Engenharia e Gestão do Conhecimento, Universidade Federal de Santa Catarina, Florianópolis, 2011.

- Scratch, 2019. Disponível em: < <https://scratch.mit.edu/>>. Acesso em: dez. 2019.
- SCRATCHED. CREATIVE COMPUTING: a design-based introduction to computational thinking. Cambridge: Harvard Graduate School Of Education, 2011.
- Sociedade Brasileira de Computação. Computação Itinerário Formativo. Disponível em:<<http://www.sbc.org.br/documentos-da-sbc/send/203-educacao-basica/1216-itinerario-formativo-da-computacao>>. Acesso em: 3 jun. 2019.
- Von Wangenheim, Christiane Gresse; Nunes, Vinícius Rodrigues; Santos, Giovane Daniel dos. Ensino de Computação com SCRATCH no Ensino Fundamental – Um Estudo de Caso. Revista Brasileira de Informática na Educação, [s.l.], v. 22, n. 03, p.115-125, 23 nov. 2014. Sociedade Brasileira de Computacao - SB. <http://dx.doi.org/10.5753/rbie.2014.22.03.115>.
- Weintrop, D.; Wilensky, U.. To Block or not to Block, That is the Question: Students' Perceptions of Blocks-based Programming. In: INTERNATIONAL CONFERENCE ON INTERACTION DESIGN AND CHILDREN, 14., 2015, Boston. Proceedings of IDC 2015: The 14th International Conference on Interaction Design and Children. Boston: Association For Computing Machinery, Inc, 2015. p. 199 - 208.
- WING, J. M.; Computational Thinking. 2006. Disponível em: <cs.cmu.edu/~15110-s13/Wing06-ct.pdf>. Acesso em 30.05.2019.

ANEXO

ANEXO A – Aprovação do Comitê de Ética em Pesquisa (CEP) da UNESC para a realização da pesquisa



RESOLUÇÃO

O Comitê de Ética em Pesquisa da UNESC, reconhecido pela Comissão Nacional de Ética em Pesquisa (CONEP)/ Ministério da Saúde analisou o projeto abaixo:

Parecer n.º: 3.889.080

CAAE: 26670919.1.0000.0119


Pesquisador(a) Responsável: LUCIANO ANTUNES

Pesquisador(a): MARCELO MILIOLI BRISTOT FILHO

Título: "SCRATCH E O ENSINO DO PENSAMENTO COMPUTACIONAL APLICADO AOS ESTUDANTES DO COLÉGIO UNESC".

Este projeto foi aprovado em seus aspectos éticos e metodológicos, de acordo com as Diretrizes e Normas Internacionais e Nacionais. Todas e qualquer alteração do Projeto deverá ser comunicada ao CEP. Os membros do CEP não participaram do processo de avaliação dos projetos onde constam como pesquisadores.

Criciúma, 28 de fevereiro de 2020.


Marco Antônio da Silva
Coordenador do CEP